

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2000年 6月 9日

出 願 番 号
Application Number: 特願2000-174573

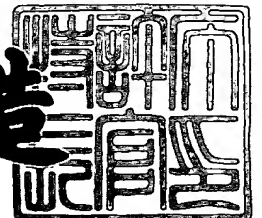
出 願 人
Applicant(s): 任天堂株式会社



2000年10月27日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3087984

【書類名】 特許願

【整理番号】 N000675

【あて先】 特許庁長官 殿

【国際特許分類】 A63F 13/00

【発明者】

【住所又は居所】 京都市東山区福稲上高松町 6 0 番地
任天堂株式会社内

【氏名】 増山 巖

【発明者】

【住所又は居所】 京都市東山区福稲上高松町 6 0 番地
任天堂株式会社内

【氏名】 鈴木 利明

【発明者】

【住所又は居所】 京都市東山区福稲上高松町 6 0 番地
任天堂株式会社内

【氏名】 田原 明

【特許出願人】

【識別番号】 000233778

【氏名又は名称】 任天堂株式会社

【代表者】 山内 博

【先の出願に基づく優先権主張】

【出願番号】 平成11年特許願第282592号

【出願日】 平成11年10月 4日

【手数料の表示】

【予納台帳番号】 056085

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ゲームシステム及びそれに用いられるゲーム情報記憶媒体

【特許請求の範囲】

【請求項 1】

ゲームプログラムを記憶するゲームプログラム記憶手段とゲームプログラムを実行する処理手段とを備えるゲーム装置に対して、処理手段の処理結果に基づく画像を表示する表示手段が関連的に設けられたゲームシステムであって、

プレイヤによって把持されるハウジング、および

前記ハウジングに関連して設けられ、かつハウジングに加えられた変化の量と方向の少なくとも一方を検出する変化状態検出手段を備え、

前記ゲームプログラム記憶手段は、

ゲームプレイ可能な空間を表示するための画像データを含むゲーム空間データと、

前記ゲーム空間データに基づいて前記表示手段にゲーム空間を表示させるための表示制御プログラムと、

前記変化状態検出手段の出力に基づいて、前記ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、前記ゲーム空間の状態を変化させるようにシミュレートするシミュレーションプログラムとを記憶したことを特徴とする、ゲームシステム。

【請求項 2】

前記変化状態検出手段は、変化量と変化方向として、前記ハウジングに加えられた傾きの傾き量と傾き方向の少なくとも一方を検出するものであり、

前記シミュレーションプログラムは、前記ハウジングに加えられた傾き量と傾き方向の少なくとも一方に関連させて、前記ゲーム空間が傾けられた状態となるようにシミュレートすることを特徴とする、請求項 1 に記載のゲームシステム。

【請求項 3】

前記変化状態検出手段は、変化量と変化方向として、前記ハウジングに加えられた運動の運動量と運動方向の少なくとも一方を検出するものであり、

前記シミュレーションプログラムは、前記ハウジングに加えられた運動量と運

動方向の少なくとも一方に関連する運動状態になるように、前記ゲーム空間をシミュレートすることを特徴とする、請求項 1 に記載のゲームシステム。

【請求項 4】

前記変化状態検出手段は、変化量と変化方向として、前記ハウジングに加えられた衝撃の衝撃量と衝撃方向の少なくとも一方を検出するものであり、

前記シミュレーションプログラムは、前記ハウジングに加えられた衝撃量と衝撃方向の少なくとも一方に関連させて、前記ゲーム空間が衝撃を加えられた状態となるようにシミュレートすることを特徴とする、請求項 1 に記載のゲームシステム。

【請求項 5】

前記変化状態検出手段は、前記ハウジングに加えられた変化の量と方向の両方を検出するものであり、

前記シミュレーションプログラムは、前記ハウジングに加えられた変化量と変化方向の両方に関連させて、前記ゲーム空間の状態を変化させるようにシミュレートすることを特徴とする、請求項 1 ないし請求項 4 のいずれかに記載のゲームシステム。

【請求項 6】

前記ハウジングは、前記ゲーム装置のハウジングであり、

前記ゲーム装置は、前記表示手段が前記ハウジングの一方主面に一体的に設けられた携帯ゲーム装置である、請求項 1 に記載のゲームシステム。

【請求項 7】

前記ゲームプログラム記憶手段は、カートリッジに収納され、かつ前記携帯ゲーム装置の前記ハウジングに対して着脱自在に装着され、

前記変化状態検出手段は、前記カートリッジに収納され、当該カートリッジが前記携帯ゲーム装置の前記ハウジングに装着されたとき、携帯ゲーム装置のハウジングに加えられた変化量と変化方向の少なくとも一方を検出する、請求項 6 に記載のゲームシステム。

【請求項 8】

前記変化状態検出手段は、前記携帯ゲーム機のハウジングに加えられる変化状

態によって道具として操作したことを検出するものであり、

前記ゲームプログラム記憶手段は、前記ゲーム空間上を移動可能な移動キャラクターを表示するためのキャラクターデータ記憶部を含み、

前記ゲーム空間データは、前記ゲーム空間上に表示される前記移動キャラクターの動きを制御する機能を有する道具を連想させる表示を行うための画像データであり、

前記ゲームプログラム記憶手段は、前記キャラクターデータ記憶部に記憶されている移動キャラクターを読み出し、前記変化状態検出手段の出力に基づいて、前記ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、移動キャラクターの表示状態が前記道具によって制御されているように処理するキャラクター制御プログラムを含む、請求項 7 に記載のゲームシステム。

【請求項 9】

前記ゲームプログラム記憶手段は、前記ゲーム空間上を移動可能な移動キャラクターを表示するためのキャラクターデータ記憶部を含み、

前記ゲームプログラム記憶手段は、前記キャラクターデータ記憶部に記憶されている移動キャラクターを読み出し、前記変化状態検出手段の出力に基づいて、前記ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、移動キャラクターの表示状態を変化させるように制御するキャラクター制御プログラムを含む、請求項 1 に記載のゲームシステム。

【請求項 1 0】

前記ゲームプログラム記憶手段は、プレイヤーの操作とは無関係にゲーム空間上を予め定められたプログラムに従って第 1 の動きをするノンプレイヤーキャラクターを表示するためのノンプレイヤーキャラクターデータ記憶部とを含み、

前記シミュレーションプログラムは、

前記変化状態検出手段によって変化量と変化方向のいずれの変化状態が検出されないとき、前記ノンプレイヤーキャラクターがプログラムによって予め定められた第 1 の動きをするように制御し、

前記変化状態検出手段によって変化量と変化方向の少なくとも一方の変化状態が検出されないとき前記ノンプレイヤーキャラクターが第 1 の動きに加えて変化

状態検出手段の出力に基づく変化量と変化方向の少なくとも一方に関連した第2の動きをするように制御することを特徴とする、請求項1に記載のゲームシステム。

【請求項11】

前記ゲームプログラム記憶手段は、前記ゲーム空間上を移動可能な移動キャラクタを表示するためのキャラクタデータ記憶部を含み、

前記ゲーム空間データは、前記移動キャラクタがゲーム空間上を移動するときに、移動キャラクタの動きが他の領域よりも異なるように定められた特定領域を表示するためのデータを含み、

前記シミュレーションプログラムは、前記変化状態検出手段の出力に基づいて前記ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、前記移動キャラクタの表示状態を制御し、移動キャラクタが前記特定領域に存在するときに、移動キャラクタの動きを他の領域に存在していたときとは異ならせるように表示制御することを特徴とする、請求項1に記載のゲームシステム。

【請求項12】

前記ゲーム空間データは、前記表示手段によって表示可能な表示領域よりも大きなゲーム空間を表示するための空間データを含み、

前記表示制御プログラムは、前記ゲーム空間のうちの表示領域の範囲内にある一部のゲーム空間の画像データを前記表示手段に表示させるものであり、

前記シミュレーションプログラムは、前記変化状態検出手段の出力の変化量と変化方向の少なくとも一方に基づいて表示領域内に存在するゲーム空間のみの状態を変化させるようにシミュレートすることを特徴とする、請求項1に記載のゲームシステム。

【請求項13】

前記変化状態検出手段は、変化量として前記ハウジングの移動量を検出し、かつ変化方向として移動方向を検出するものであり、

前記ゲーム空間データは、前記表示手段の表示領域よりも大きなゲーム空間を表示するための空間データを含み、

前記表示制御プログラムは、表示領域に対応するゲーム空間の一部の空間領域

を前記表示手段に表示させ、前記ハウジングの移動に応じて、ゲーム空間の表示領域を移動方向へ移動量に相当する領域だけ徐々に移動させることを特徴とする、請求項 1 に記載のゲームシステム。

【請求項 1 4】

前記ゲーム装置は、プレイヤによって操作される操作手段を前記ハウジングの一方主面に備え、

前記シミュレーションプログラムは、前記変化状態検出手段の検出出力と前記操作手段の操作状態に基づいて、ゲーム空間の状態をシミュレートするように変化させることを特徴とする、請求項 1 に記載のゲームシステム。

【請求項 1 5】

前記ゲームプログラム記憶手段は、前記シミュレーションプログラムに基づく前記ゲーム空間の状態に対応した音を発生する音発生プログラムを含む、請求項 1 に記載のゲームシステム。

【請求項 1 6】

表示手段が関連的に設けられ、プレイヤによって把持されるハウジングを含む操作手段と、ハウジングに関連して設けられかつハウジングに加えられた変化の量と方向の少なくとも一方を検出する変化状態検出手段と、プログラムを処理することによって得られる画像を表示手段に表示させる処理手段とから構成されるゲームシステムに対して、着脱自在に装着されかつゲームプログラムを記憶するゲーム情報記憶媒体であって、

ゲームプレイ可能な空間を表示するための画像データを含むゲーム空間データと、

前記ゲーム空間データに基づいて前記表示手段にゲーム空間を表示させるための表示制御プログラムと、

前記変化状態検出手段の出力に基づいて、前記ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、前記ゲーム空間の状態を変化させるようにシミュレートするシミュレーションプログラムとを記憶したことを特徴とする、ゲーム情報記憶媒体。

【請求項 1 7】

表示手段が一体的に設けられかつプレイヤによって把持されるハウジングと、プログラムを処理することによって得られる画像を表示手段に表示させる処理手段とを含む携帯ゲーム装置に対して、着脱自在に装着されかつゲームプログラムを記憶するゲーム情報記憶媒体であって、

前記ゲーム情報記憶媒体に関連して設けられ、かつ前記携帯ゲーム装置のハウジングに加えられた変化の量と方向の少なくとも一方を検出する変化状態検出手段を備え、

ゲームプレイ可能な空間を表示するための画像データを含むゲーム空間データと、

前記ゲーム空間データに基づいて前記表示手段にゲーム空間を表示させるための表示制御プログラムと、

前記変化状態検出手段の出力に基づいて、前記ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、前記ゲーム空間の状態を変化させるようにシミュレートするシミュレーションプログラムとを記憶したことを特徴とする、ゲーム情報記憶媒体。

【請求項 1 8】

前記変化状態検出手段は、前記ハウジングに加えられた変化の量と方向の両方を検出するものであり、

前記シミュレーションプログラムは、前記ハウジングに加えられた変化量と変化方向の両方に関連させて、前記ゲーム空間の状態を変化させるようにシミュレートすることを特徴とする、請求項 1 7 に記載のゲーム情報記憶媒体。

【請求項 1 9】

少なくとも 2 台のゲーム装置によって構成されるゲームシステムであって、

前記 2 台のゲーム装置は、それぞれゲームプログラムを記憶するゲームプログラム記憶手段と、ゲームプログラムを実行する処理手段と、プレイヤによって把持されるハウジングとを備え、処理手段の処理結果に基づく画像を表示する表示手段が関連的に設けられ、

前記 2 台のゲーム装置の少なくとも一方は、前記ハウジングに関連して設けられ、かつハウジングに加えられた変化量と変化方向の少なくとも一方を検出する

変化状態検出手段を備え、

前記 2 台のゲーム装置に接続されかつ相互に関連するデータを相手側のゲーム装置に伝送するデータ伝送手段をさらに備え、

前記 2 台のゲーム装置のそれぞれの前記ゲームプログラム記憶手段は、

ゲームプレイ可能な空間を表示するための画像データを含むゲーム空間データと、

前記ゲーム空間データに基づいて前記表示手段にゲーム空間を表示させるための表示制御プログラムとを記憶し、

前記 2 台のゲーム装置の少なくとも他方の前記ゲームプログラム記憶手段は、前記データ伝送手段を介して伝送される前記一方のゲーム装置の前記変化状態検出手段の出力に基づいて、一方のゲーム装置の前記ハウジングに加えられた変化量と変化方向の少なくとも 1 つに関連させて、他方のゲーム装置の前記ゲーム空間の状態を変化させるようにシミュレートするシミュレーションプログラムをさらに記憶したことを特徴とする、ゲームシステム。

【請求項 2 0】

前記変化状態検出手段は、2 台のゲーム装置のそれぞれに備えられ、

前記 2 台のゲーム装置のそれぞれの前記ゲームプログラム記憶手段は、一方のゲーム装置の前記変化状態検出手段の出力に基づいて、一方のゲーム装置の前記ハウジングに加えられた変化量と変化方向の少なくとも 1 つに関連させて、他方のゲーム装置の前記ゲーム空間の状態を変化させるようにシミュレートするシミュレーションプログラムを記憶したことを特徴とする、請求項 1 9 に記載のゲームシステム。

【請求項 2 1】

前記一方のゲーム装置の前記ゲームプログラム記憶手段に記憶されるゲーム空間データと、前記他方のゲーム装置の前記ゲームプログラム記憶手段に記憶されるゲーム空間データとは、同一のゲーム空間データに選ばれ、

前記一方のゲーム装置の前記シミュレーションプログラムは、当該ゲーム空間制御プログラムによってシミュレートされる他方のゲーム空間の状態に対応して、前記一方のゲーム装置の前記ゲーム空間の状態を変化させ、

前記他方のゲーム装置の前記シミュレーションプログラムは、当該ゲーム空間制御プログラムによってシミュレートされる一方のゲーム空間の状態に対応して、前記他方のゲーム装置の前記ゲーム空間の状態を変化させることを特徴とする、請求項 1 9 に記載のゲームシステム。

【発明の詳細な説明】

【0 0 0 1】

【産業上の利用分野】

この発明はゲームシステム及びそれに用いられるゲーム情報記憶媒体に関し、特に携帯ゲーム装置のハウジング又はビデオゲーム装置のコントローラに加えられる傾き、運動又は衝撃等の変化量・変化方向を検出して操作情報としてゲーム空間の状態を変化させる、ゲームシステム及びそれに用いられるゲーム情報記憶媒体に関する。

【0 0 0 2】

【従来技術】

従来のゲーム装置における操作は、プレイヤーがビデオゲーム機のコントローラ（操作器のハウジング）又は携帯型ゲーム装置のハウジングを両手で把持しつつ、方向指示キー（ジョイスティック）やボタン等の操作スイッチを操作することによって行われていた。例えば、プレイヤーが方向指示キーの上下左右の何れか 1 つの押点を押圧すると、移動（プレイヤー）キャラクターが上下左右の何れか 1 つの押圧方向に移動し、動作ボタンを操作すると移動キャラクターがジャンプ等の動作ボタンに定義付けされた動作をするように、移動キャラクターの表示状態が変化する。

また、従来のゲーム装置又はゲームソフト（ゲーム情報記憶媒体）は、プレイヤーが操作スイッチを操作してプレイヤーの分身である移動（プレイヤー）キャラクターの画面上の動きを変化させるものであるため、ゲーム空間（又は背景画面）をプレイヤーの操作によって自由に変更できるものではなかった。

【0 0 0 3】

【発明が解決しようとする課題】

従来のゲーム操作方法は、プレイヤーがゲームソフトの取扱説明書等に表示された

ゲーム操作方法を覚えることが必要であり、汎用的な操作スイッチを使用するものであるため、プレイヤーによる操作スイッチの操作感覚と一致したゲーム空間（又はゲーム画面）の変化を実現できず、操作感覚と画面の表示状態とが対応しない場合もあった。そのため、プレイヤーはゲーム操作方法を習得するまでは、ゲームプレイに集中できず、興趣を損なうこともあった。

また、従来のゲーム装置又はゲームソフトは、プレイヤーの操作に応じてゲーム空間（又は背景画面）を変化させることができず、ゲーム空間の変化に乏しく、面白味に欠ける場合もあった。

【0004】

それゆえに、この発明の主たる目的は、ゲーム空間の状態をプレイヤーの操作によって変化させることが可能な、ゲームシステム及びそれに用いられるゲーム情報記憶媒体を提供することである。

この発明の他の目的は、簡単な操作でゲーム空間の状態を変化させることが可能であり、プレイヤーが操作方法に熟練する必要がなく、ゲームプレイに集中できるとともに、熱中度を高められる、ゲームシステム及びそれに用いられるゲーム情報記憶媒体を提供することである。

この発明のその他の目的は、プレイヤーの操作とゲーム空間の状態の変化を対応させて、操作感覚と対応したゲーム画面の変化を実現し得る、ゲームシステム及びそれ用いられるゲーム情報記憶媒体を提供することである。

この発明のさらに他の目的は、複数のゲーム装置と連動してゲーム空間の状態を変化でき、複数のプレイヤーによる協力又は対戦によってゲーム空間の変化状態の種類を豊富にでき、ゲームの興趣を高められて、仮想現実感を味わうことができる、ゲームシステム及びそれ用いられるゲーム情報記憶媒体を提供することである。

【0005】

【課題を解決するための手段】

第1の発明（請求項1に係る発明）は、ゲームプログラムを記憶するゲームプログラム記憶手段とゲームプログラムを実行する処理手段とを備えるゲーム装置に対して、処理手段の処理結果に基づく画像を表示する表示手段が関連的に設け

られたゲームシステムである。そして、プレイヤによって把持されるハウジング、及び変化状態検出手段を備える。変化状態検出手段は、ハウジングに関連して設けられ、かつハウジングに加えられた変化の量（例えば、傾き量、運動量、衝撃量等）と変化方向（例えば、傾き方向、運動方向、衝撃方向等）の少なくとも一方を検出する。ゲームプログラム記憶手段は、ゲーム空間データと、表示制御プログラムと、シミュレーションプログラムとを記憶する。

ゲーム空間データは、ゲームプレイ可能な空間を表示するための画像データを含む。表示制御プログラムは、ゲーム空間データに基づいて表示手段にゲーム空間を表示させる。シミュレーションプログラムは、状態検出手段の出力に基づいて、ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、ゲーム空間の状態を変化させるようにシミュレートする。

【0006】

ここで、ゲーム空間とは、プレイ可能なゲーム中の世界を意味するものであり、ゲームの種類又はジャンルによって異なり、表示画面を通じてプレイヤに示されるものである。例えば、移動（又はプレイヤ）キャラクターがその中を移動するアクションゲームやロールプレイングゲームでは背景、迷路その他のマップ、格闘ゲームではリング（それに加えて観客席やリング上部の空間も含む）、レースゲームでは走行可能なコースやコースの周辺の空間、シューティングゲームではキャラクターの背景となるような宇宙空間等の背景画面（但し、キャラクターは必須ではなく、キャラクターが存在しないゲーム空間も考えられ得る）、道具を使用したゲームでは道具の使用を連想させるような画面等がゲーム空間となる。

シミュレートとは、ハウジングに加えられた傾き、運動又は衝撃の量的なものと方向性の少なくとも一方に基づいて、実空間上に生じる変化をゲーム空間の状態変化として擬似的に表現するゲーム制御のことである。ゲーム制御には、ゲーム空間そのものの状態変化をシミュレートする場合と、ゲーム空間の状態が変化することによって他の物体への間接的な影響をシミュレートする場合を含む。前者は、ハウジングに衝撃が与えられたとき、ゲーム空間にエネルギーが与えられたものと想定してゲーム空間の地形を変形させるようにシミュレートする場合である。後者は、ハウジングが傾けられたとき、ゲーム空間の一例の迷路板が傾い

たものと仮定として、迷路板の上に存在するボールが転がるようにシミュレートする場合である。ゲーム空間そのものの状態変化をシミュレートする場合は、地形変形を引き起こすもののような表示上の変化を引き起こすものの他に、ゲーム空間の温度上昇等のパラメータが変化するものも考えられる。

【 0 0 0 7 】

第 2 の発明（請求項 1 6 に係る発明）は、表示手段が関連的に設けられ、プレイヤによって把持されるハウジングを含む操作手段と、ハウジングに関連して設けられかつハウジングに加えられた変化の量と方向の少なくとも一方を検出する変化状態検出手段と、プログラムを処理することによって得られる画像を表示手段に表示させる処理手段とから構成されるゲームシステムに対して、着脱自在に装着されかつゲームプログラムを記憶するゲーム情報記憶媒体である。ゲーム情報記憶媒体は、ゲーム空間データと、表示制御プログラムと、シミュレーションプログラムとを記憶する。

ゲーム空間データは、ゲームプレイ可能な空間を表示するための画像データを含む。表示制御プログラムは、ゲーム空間データに基づいて表示手段にゲーム空間を表示させるものである。シミュレーションプログラムは、変化状態検出手段の出力に基づいて、ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、ゲーム空間の状態を変化させるようにシミュレートするものである。

【 0 0 0 8 】

第 3 の発明（請求項 1 7 に係る発明）は、表示手段が一体的に設けられかつプレイヤによって把持されるハウジングと、プログラムを処理することによって得られる画像を表示手段に表示させる処理手段とを含む携帯ゲーム装置に対して、着脱自在に装着されかつゲームプログラムを記憶するゲーム情報記憶媒体であり、ゲーム情報記憶媒体に関連して設けられ、かつ携帯ゲーム装置のハウジングに加えられた変化の量と方向の少なくとも一方を検出する変化状態検出手段を備える。

ゲーム情報記憶媒体は、ゲーム空間データと、表示制御プログラムと、シミュレーションプログラムとを記憶する。ゲーム空間データは、ゲームプレイ可能な

空間を表示するための画像データを含む。表示制御プログラムは、ゲーム空間データに基づいて前記表示手段にゲーム空間を表示させるプログラムである。シミュレーションプログラムは、変化状態検出手段の出力に基づいて、ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、ゲーム空間の状態を変化させるようにシミュレートするプログラムである。

【0009】

第4の発明（請求項19に係る発明）は、少なくとも2台のゲーム装置によって構成されるゲームシステムであって、複数のゲーム装置の連動を行うものである。2台のゲーム装置は、それぞれゲームプログラムを記憶するゲームプログラム記憶手段と、ゲームプログラムを実行する処理手段と、プレイヤによって把持されるハウジングとを備え、処理手段の処理結果に基づく画像を表示する表示手段が関連的に設けられる。2台のゲーム装置の少なくとも一方は、ハウジングに関連して設けられ、かつハウジングに加えられた変化量と変化方向の少なくとも一方を検出する変化状態検出手段を備える。2台のゲーム装置に接続されかつ相互に関連するデータを相手側のゲーム装置に伝送するデータ伝送手段をさらに備える。

2台のゲーム装置のそれぞれのゲームプログラム記憶手段は、ゲーム空間データと、表示制御プログラムとを記憶する。ゲーム空間データは、ゲームプレイ可能な空間を表示するための画像データを含む。表示制御プログラムは、ゲーム空間データに基づいて表示手段にゲーム空間を表示させるプログラムである。2台のゲーム装置の少なくとも他方のゲームプログラム記憶手段は、データ伝送手段を介して伝送される一方のゲーム装置の変化状態検出手段の出力に基づいて、一方のゲーム装置の前記ハウジングに加えられた変化量と変化方向の少なくとも1つに関連させて、他方のゲーム装置の前記ゲーム空間の状態を変化させるようにシミュレートするシミュレーションプログラムをさらに記憶する。

【0010】

【発明の効果】

この発明によれば、ゲーム空間の状態を変化させることが可能な、ゲームシステム及びそれ用いられるゲーム情報記憶媒体が得られる。

また、この発明によれば、簡単な操作でゲーム空間の状態を変化させることができ、プレイヤーが操作方法に熟練する必要がなく、ゲームプレイに集中でき、熱中度を高められる、ゲームシステム及びそれに用いられるゲーム情報記憶媒体が得られる。

また、この発明によれば、プレイヤーの操作とゲーム空間の状態の変化を対応させて、操作感覚と対応したゲーム画面の変化を実現することができる、ゲームシステム及びそれ用いられるゲーム情報記憶媒体が得られる。

さらに、この発明によれば、複数のゲーム装置と連動してゲーム空間の状態を変化でき、複数のプレイヤーによる協力又は対戦によってゲーム空間の変化状態の種類を豊富にでき、ゲームの興趣を高められて、仮想現実感を味わうことができる、ゲームシステムが得られる。

【 0 0 1 1 】

【発明の実施の形態】

（第 1 実施例）

図 1 ～図 4 0 を参照して、本発明の第 1 の実施例の携帯型ゲーム装置を説明する。図 1 は携帯型ゲーム装置（以下、単に「ゲーム装置」と呼ぶ）の外観図である。ゲーム装置は、ゲーム装置本体 1 0 とゲーム装置本体 1 0 に着脱自在なゲームカートリッジ（以下「カートリッジ」と略称する）3 0 とから構成される。カートリッジ 3 0 は、ゲーム装置本体 1 0 に装着されたときに電氣的に接続される。ゲーム装置本体 1 0 は、ハウジング 1 1 を備え、ハウジング 1 1 の内部に後述する図 3 に示すように回路構成される基板を含む。ハウジング 1 1 の一方主面には LCD 1 2 および操作スイッチ 1 3 a ～ 1 3 e が設けられ、他方主面にはカートリッジ 3 0 を装着するための孔 1 4 が形成される。また側面には必要に応じて他のゲーム装置と通信するための通信ケーブルを接続するコネクタ 1 5 が設けられる。

【 0 0 1 2 】

図 2 は、ゲーム装置と X Y Z 軸との関係を示した図である。LCD 1 2 を上方に向けて操作スイッチ部が手前になるようにゲーム装置を配置したとき、ゲーム装置の横方向を X 軸（右方向がプラス方向）とし、縦方向を Y 軸（奥方向がプラ

ス方向)とし、厚み方向をZ軸(上方向がプラス方向)とする。

【0013】

図3はゲーム装置のブロック図である。ゲーム装置本体10は、基板27を内蔵し、基板27にはCPU21が実装される。CPU21には、LCDドライバ22と操作スイッチ13とサウンド発生回路23と通信用インターフェース24と表示用RAM25とワークRAM26とが接続される。サウンド発生回路23には、スピーカー16が接続される。通信用インターフェース24は、コネクタ15および通信ケーブル50を介して他のゲーム装置40と接続される。なお、他のゲーム装置40との通信の方法は通信ケーブル50による方法を図示したが、無線や携帯電話等を使用したものでも良い。

【0014】

カートリッジ30は、基板36を内蔵し、基板36には図16を参照して後述するようなゲームプログラムおよびゲームデータを記憶したプログラムROM34と、図19を参照して後述するようなゲームデータを記憶するバックアップRAM35が実装される。カートリッジ30は、これらの記憶手段に加えて、ゲーム装置本体の傾き、運動および衝撃を検出する検出手段の一例として、X軸方向およびY軸方向の加速度を検出するXY軸加速度センサ31と、Z軸方向の加速度を検出するZ軸接点スイッチ32を含む。また、カートリッジ30は、加速度検出手段のインターフェースであるセンサインターフェース33を含む。X軸、Y軸、Z軸方向すべての加速度を検出可能な3軸の加速度センサを使用する場合、Z軸接点スイッチ32が不要となるが、2軸加速度センサ(XY軸加速度センサ)の方が安価であり、本実施例ではZ軸方向の加速度検出は高い精度を必要としないので、構造が簡単で安価なZ軸接点スイッチ32を用いることにする。また、XY軸方向についても高い精度を必要としない場合は、Z軸接点スイッチと同じような構造の検出手段をXY軸方向の加速度検出に使用しても良い。

【0015】

プログラムROM34に記憶されたゲームプログラムはCPU21によって実行される。ゲームプログラムの実行の際に必要な一時的なデータがワークRAM26に記憶される。ゲーム装置の電源をオフしたときにも持続的に記憶すべきゲ

ームデータがバックアップRAM 35に記憶される。CPU 21がゲームプログラムを実行して得た表示データが表示用RAM 25に記憶され、LCDドライバ22を介してLCD 12に表示される。同様に、CPU 21がゲームプログラムを実行して得たサウンドデータがサウンド発生回路23に送られてスピーカー16から効果音としてサウンドが発生される。操作スイッチ13はゲーム操作をするためのものであるが、本実施例においては操作スイッチ13は補助的なものである。プレイヤは、主として、ゲーム装置を傾けたり、運動させたり、衝撃を与えたりしてゲーム操作をおこなう。このゲーム装置の傾き、運動、衝撃のゲーム操作がXY軸加速度センサ31およびZ軸接点スイッチ32によって検出される。これらの加速度検出手段の出力値を利用して、CPU 21がゲームプログラムを実行する。

【0016】

複数のゲーム装置を使用するゲームの場合、CPU 21がゲームプログラムを実行して得たデータは通信用インターフェース24に送られてコネクタ15および通信ケーブル50を介して他のゲーム装置40に送られる。また、他のゲーム装置40のデータが通信ケーブル50、コネクタ15および通信用インターフェース24を介してCPU 21に送られる。

【0017】

図4はセンサインターフェース33の詳細ブロック図である。センサインターフェース33は、Xカウンタ331、Yカウンタ332、カウント停止回路333、ラッチ334と335、デコーダ336および汎用I/Oポート337を含む。Xカウンタ331は、XY軸加速度センサ31のX軸出力に基づいてクロック信号Φのパルスをカウントする。Yカウンタ332は、Y軸出力に基づいてクロック信号Φのパルスをカウントする。カウント停止回路333は、XY軸加速度センサ31のX軸出力の立ち下がりに応答して、Xカウンタ331にカウント停止信号を送り、Y軸出力の立ち下がりに応答してYカウンタ332にカウント停止信号を送る。ラッチ334、335は、Xカウンタ331、Yカウンタ332の値をそれぞれ保持する。デコーダ336は、Xカウンタ331とYカウンタ332とラッチ334とラッチ335にスタート/リセット信号を送信する。汎

用 I / O ポート 3 3 7 は、拡張ユニットを接続するために用いられる。ラッチ 3 3 4, 3 3 5 は Z 軸接点スイッチの出力値 (0 または 1) も保持する。具体的にはラッチ 3 3 4, 3 3 5 の最上位ビットが Z 軸接点スイッチの出力値に割り当てられ、残りの下位ビットが X カウンタまたは Y カウンタの値に割り当てられる。汎用 I / O ポート 3 3 7 に接続される拡張ユニットは、例えばゲームに現実感を与えるためにゲームプログラムに連動して振動するような振動ユニット等がある。

【 0 0 1 8 】

図 5 はセンサインターフェース 3 3 が X Y 軸加速度センサ 3 1 の出力から加速度の大きさに応じたカウント値を計測する原理を示した図である。本実施例における X Y 軸加速度センサ 3 1 は、波形の 1 周期 (期間 1) のうちのデューティ比を変化させることによって加速度の大きさを表す信号を出力する。この場合、1 周期のうちのハイレベル期間 (期間 2 または期間 3) の比率が大きいほど大きな加速度を検出したことを示す。また、X Y 軸加速度センサ 3 1 は、X 軸出力から X 軸方向の加速度の大きさを出力し、Y 軸出力から Y 軸方向の加速度の大きさを出力する。

【 0 0 1 9 】

X カウンタ 3 3 1 は、デコーダ 3 3 6 から出力されるカウントスタート信号がローレベルになると、X 軸出力のローレベルからハイレベルへの立ち上がりを検出した後、カウント動作を開始する。具体的には、X カウンタ 3 3 1 はクロック信号 Φ が与えられる毎にそのカウント値を歩進し、カウント停止回路 3 3 3 からのカウント停止信号に応じてカウント動作を停止する。このようにして、X カウンタ 3 3 1 はカウントスタート信号がローレベルになった直後の X 軸出力のハイレベルへの立ち上がりからローレベルに立ち下がるまでの期間 (期間 2) 中に、クロック信号 Φ をカウントする。Y カウンタ 3 3 2 も同様に、カウントスタート信号がローレベルになった直後の Y 軸出力のハイレベルへの立ち上がりからローレベルへの立ち下がりまでの期間 (期間 3) 中に、クロック信号 Φ をカウントする。このようにして、X カウンタ 3 3 1 は X 軸方向の加速度の大きさに応じたカウント値を保持し、Y カウンタ 3 3 2 は Y 軸方向の加速度の大きさに応じたカウ

ント値を保持する。Xカウンタ331およびYカウンタ332の値はラッチ334またはラッチ335に保持され、ラッチ334およびラッチ335のデータは、データバスを介してCPU21に読み出されてゲームプログラムで利用される。

【0020】

Xカウンタ331およびYカウンタ332は、例えば、0から31までカウントし、例えば、カウント値15を基準（加速度0）として、カウント値が0のときが-2G（マイナス方向に重力加速度の2倍）、カウント値が31のときが2G（プラス方向に重力加速度の2倍）であるように設定される。CPU21は、ゲームプログラムに基づいてこれらのカウント値を読み込むが、このとき、カウント値15を0、カウント値0を-15、カウント値31を16として読み込むので、XY軸加速度センサ31がマイナス方向の加速度を検出したときはCPUの読み込み値はマイナスであり、プラス方向の加速度を検出したときにはCPUの読み込み値はプラスとなる。

【0021】

図6はZ軸接点スイッチ32の構造図である。Z軸接点スイッチ32は導体よりなる球接点321と接点322と接点323と箱体324から構成される。具体的には、球接点321は、箱体324の空間内のほぼ中央部で移動自在に支持される（箱体324の内面底部には球接点321をほぼ中央に支持するためのくぼみ（324a）が設けられる）。箱体324の上部には、それぞれの一方端部に半円状の切欠部（322a, 323a）を有する板状の接点322と接点323とが一方端部を対向させた状態で、他方端部が基板36に固着される。また、箱体324は、接点322および接点323によって吊り下げられた状態で、基板36に固定的に保持される。このような構成によって、カートリッジ30がZ軸方向（プラス方向またはマイナス方向）へ勢いよく移動されると、図7に示すように球接点321が箱体324内でZ軸方向へ移動して接点322と接点323にほぼ同時に接触して、接点322と接点323が球接点321を介して導通状態となり、Z軸方向への加速度入力のあったことが検出される。接点322と接点323の導通時間に基づいてZ軸方向への加速度の大小が検出されることに

なる。なお、カートリッジ 3 0 が穏やかに傾けられた場合は、球接点 3 2 1 が箱体 3 2 4 内を移動するが接点 3 2 2 と接点 3 2 3 を短絡しないので、加速度を検出しない。

【 0 0 2 2 】

図 8 はゲーム画面の一例である。ゲーム画面には、プレイヤーキャラクタの一例のボール 6 1 と、敵キャラクタ（以下「NPC」と略す）の一例の亀 6 2 と、迷路を構成する壁 6 3 や穴 6 4 とが表示される。ゲームマップは、LCD 1 2 の表示範囲より広い仮想マップであるので、LCD 1 2 には、ゲームマップの一部領域のみが表示され、プレイヤーキャラクタの移動に伴ってスクロールする。また、LCD 1 2 には亀 6 2 a ～ 6 2 c の 3 匹のみ表示されているが、ゲームマップ上には他にも多数の亀が存在する。また、ゲームマップには、床面と氷面と水中等の地形が存在する。

【 0 0 2 3 】

ボール 6 1 は、プレイヤーがゲーム装置を傾けたり、運動や衝撃を与えるように操作することによって、その移動量や移動方向が変更され、必要に応じてその形状も変化される。亀 6 2 は、ゲームプログラムによって移動制御（自律移動）されるが、プレイヤーがゲーム装置を傾けたり、運動や衝撃を与えた場合にも移動したり表示変化する。

【 0 0 2 4 】

このゲームの概要を説明すると、プレイヤーは壁 6 3 によって迷路状になっているゲームマップの中でボール 6 1 を操作して、NPC である亀 6 2 a ～ 6 2 c をボール 6 1 で押しつぶす。押しつぶされた亀は消滅される。ゲームマップ上の全ての亀を消滅させることに成功するとゲームクリアとなる。ゲームマップ上にはいくつかの穴 6 4 があり、この穴 6 4 にボール 6 1 が落ちると 1 回のミスとなるか、またはゲームオーバーとなる。

【 0 0 2 5 】

図 9 から図 1 2 はゲーム操作の例を示した図である。図 9 は X 軸または Y 軸方向のスライド入力を示した図である。X 軸方向の移動（スライド）が X Y 軸加速度センサ 3 1 の X 軸出力に基づいて検出され、Y 軸方向の移動（スライド）が X

Y軸加速度センサ31のY軸出力に基づいて検出される。(X軸またはY軸方向に移動させると、加速度が生じる)。図10はX軸またはY軸を中心とした傾き入力を示した図である。X軸を中心とした傾きがXY軸加速度センサ31のY軸出力に基づいて検出され、Y軸を中心として傾きがXY軸加速度センサ31のX軸出力に基づいて検出される(X軸を中心として傾きが生じると、重力によってY軸方向に加速度が生じ、Y軸を中心として傾きが生じると、重力によってX軸方向に加速度が生じる)。図11はX軸方向またはY軸方向の衝撃入力を示した図である。X軸方向の加速度入力がXY軸加速度センサ31のX軸出力から出力されるが、この出力値が一定値以上の場合、衝撃入力があったとする。また、Y軸方向の加速度入力がXY軸加速度センサ31のY軸出力から出力されるが、この出力値が一定値以上の場合、衝撃入力があったとする。図12はZ軸方向の運動入力(または衝撃入力)を示した図である。Z軸方向の運動(または衝撃)がZ軸接点スイッチ32によって検出される。

【0026】

図13乃至図15は前述のゲーム操作それぞれについて利用方法の例を示した図である。図13はスライド入力の利用方法を示した図である(図30を参照して後述するゲームマップ選択処理におけるゲーム画面の一例でもある)。LCD12の表示範囲より大きな仮想マップの一部領域をLCD12に表示している場合に、スライド入力することによって、表示領域をスクロールさせる。具体的には、X軸の+方向にスライド入力した場合は、現在の表示領域からX軸の+方向に移動した領域を表示するようにする。Y軸方向のスライド入力も同様に処理する。スライド入力をこのように処理することによって、あたかもLCD12を通して広い世界の一部を覗いているような感覚をプレイヤーに与えることができる。なお、本実施例においては、このスライド入力は、図30を参照して後述するゲームマップ選択処理において利用するのみであり、ゲームのメインの処理におけるゲームマップのスクロール処理においては、スライド入力は利用しない。ゲームマップのスクロール処理の方法については、図38乃至図40を参照して後述する。

【0027】

図 1 4 は X 軸または Y 軸を中心とした傾き入力の利用方法を示した図である。X 軸を中心とした傾き入力があったときは、ゲーム画面上のゲームキャラクター（プレイヤーキャラクター 6 1 および NPC 6 2）が Y 軸方向に平行移動するような表示をする（X 軸を中心としてプラス方向に傾いている場合には、ゲームキャラクターが Y 軸のマイナス方向に平行移動するような表示をする）。また、Y 軸を中心とした傾き入力があったときは、ゲーム画面上のゲームキャラクター（プレイヤーキャラクター 6 1 および NPC 6 2）が X 軸方向に平行移動するような表示をする（Y 軸を中心としてマイナス方向に傾いている場合には、ゲームキャラクターが X 軸のマイナス方向に平行移動するような表示をする）。傾き入力をこのように処理することによって、ゲーム空間である迷路板がゲーム装置と同じように傾いて、傾いた迷路板上をゲームキャラクターが滑っている（転がっている）かのような感覚をプレイヤーに与えることができる。なお、ゲームマップには、床面、氷面、水中等のように、ボール 6 1 の移動量を変化させる要因となる地形が混在しており、ゲームキャラクターがどこに存在しているかによって傾き入力に応じた移動量が変わる。例えば、氷面の場合は滑りやすいので移動量が大きく、水中の場合は移動量が少ないように、ボール 6 1 の制御の大きさを変化させる。

【 0 0 2 8 】

図 1 5 は衝撃入力または Z 軸方向の運動入力の利用方法を示した図である。X 軸方向または Y 軸方向の衝撃入力があったとき、傾き入力の処理（迷路板の傾きによるゲームキャラクターの移動）とは異なった処理を行う。例えば、ゲーム空間である水に波を起こす。X 軸のプラス方向の衝撃入力があった場合、X 軸のプラス方向に波を発生する。X 軸のマイナス方向の衝撃入力があった場合、X 軸のマイナス方向に波を発生する。Y 軸方向の衝撃入力についても同様である。また、X 軸方向の加速度入力を X 軸方向のベクトル成分とし、Y 軸方向の加速度入力を Y 軸方向のベクトル成分として合成したベクトルの方向に波を発生しても良い。この波によってゲームキャラクターが流されて移動するような表示をする。波に流されている間は、ゲームキャラクターが制御不可能であるようにしてもよい。また、Z 軸方向の運動入力（または衝撃入力）があったとき、プレイヤーキャラクターであるボール 6 1 がジャンプするように表示変化される。Z 軸方向の運動入力をこ

のように処理することによって、ゲーム空間である迷路板がゲーム装置と同じようにZ軸方向に運動して、迷路板上のゲームキャラクタをジャンプさせるかのような感覚をプレイヤに与えることができる。ジャンプしている間は、傾き入力があったときでもボール61が移動しない。また、Z軸方向の運動入力（または衝撃入力）があったとき、NPCである亀62は裏返る（裏返っていた亀は表向きに戻る）。亀は裏返ると滑りやすくなり、表向きの場合に比べて傾き入力があったときの移動量が大きいように移動処理される。

【0029】

図16は、プログラムROM34のメモリマップである。プログラムROM34には、CPU21によって実行されるゲームプログラムおよびゲームデータが記憶される。プログラムROM34は、具体的には、オブジェクトキャラクタデータ記憶領域34a、マップデータ記憶領域34b、加速度センサ出力値変換テーブル記憶領域34cおよびゲームプログラム記憶領域34eを含む。オブジェクトキャラクタデータ記憶領域34aには、オブジェクトキャラクタのグラフィックデータが記憶される。オブジェクトキャラクタはいくつかのポーズを持っているので（例えばNPCである亀の「表向き」と「裏向き」）、一つのキャラクタ毎にポーズに応じた複数のグラフィックデータが記憶されている。マップデータ記憶領域34bには、ゲームマップ毎のマップデータおよびゲームマップ選択用マップが記憶されている。ゲームマップ選択用マップは、図30を参照して後述するゲームマップ選択処理においてLCD12に表示される仮想マップのデータである。

【0030】

加速度センサ出力値変換テーブル記憶領域34cには、XY軸加速度センサ31およびZ軸接点スイッチ32の出力値を変換してゲームプログラムで利用するための変換テーブルが記憶されている。変換テーブルには、ゲームマップ選択処理用テーブルとプレイヤキャラクタ移動用テーブルとNPC移動用テーブルがある。また、プレイヤーキャラクタ移動用テーブルには、空中用、床面用、氷面用および水中用のテーブルがあり、プレイヤキャラクタが存在する座標の地形に応じて選択される。NPC移動用テーブルには表向き用と裏向き用のテーブルがあ

る。NPCである亀は表向きの状態と裏向きの状態があり、この状態に応じてテーブルが選択される。これらのテーブルの詳細については、図20から図26を参照して後述する。

【0031】

ゲームプログラム記憶領域34eには、CPU21によって実行されるゲームプログラムが記憶される。具体的には、図27を参照して後述するメインプログラム、図28を参照して後述するOG設定プログラム、図29を参照して後述するニュートラルポジション設定プログラム、図30を参照して後述するゲームマップ選択プログラム、図31を参照して後述するセンサ出力読取プログラム、図32から図36を参照して後述するオブジェクト移動プログラム、図37を参照して後述する衝突プログラム、図40を参照して後述する画面スクロールプログラムの他、NPC自律移動プログラムやその他のプログラムが記憶される。

【0032】

図17は、ワークRAM26のメモリマップである。ワークRAM26には、CPU21がゲームプログラムを実行する際の一時的なデータが記憶される。具体的には、ニュートラルポジションデータ記憶領域26a、加速度センサ出力値記憶領域26b、衝撃入力フラグ記憶領域26c、マップ選択画面のカメラ座標記憶領域26e、ゲームマップナンバー記憶領域26fおよびキャラクタデータ記憶領域26gが含まれる。

【0033】

ニュートラルポジションデータ記憶領域26aには、図29を参照して後述するニュートラルポジション設定処理において設定されるニュートラルポジションデータ(NPx, NPy, NPz)が記憶される。これは、ゲームプレイをするときのゲーム装置の基準の傾きに関するデータである。

【0034】

加速度センサ出力値記憶領域26bには、XY軸加速度センサ31とZ軸接点スイッチ32によって検出され、センサインターフェース33を介して図31のセンサ出力読取処理において読み出される加速度センサ出力値(INx, INy, INz)が記憶される。衝撃入力フラグ記憶領域26cには、X軸方向の加速

度入力をX軸方向のベクトル成分としY軸方向の加速度入力をY軸方向のベクトル成分として合成したベクトルの大きさが一定値以上であったときに1となる衝撃入力フラグ（FS）が記憶される。衝撃入力の判定は、図31のセンサ出力読取処理でおこなわれる。

【0035】

マップ選択画面のカメラ座標記憶領域26eには、図30を参照して後述するゲームマップ選択処理におけるゲームマップ選択用マップのうちLCD12に表示される領域の左上隅の座標（Cx, Cy）が記憶される。ゲームマップナンバー記憶領域26fには、図30を参照して後述するゲーム選択処理において、プレイヤーが選択したゲームマップに対応したナンバーデータ（MN）が記憶される。

【0036】

キャラクターデータ記憶領域26gには、キャラクター（プレイヤーキャラクターおよびNPC）毎に、移動加速度データ（Ax, Ay, Az）、移動加速度の変化量データ（dAx, dAy, dAz）、速度データ（Vx, Vy, Vz）、座標データ（X, Y, Z）、前回の座標データ（Px, Py, Pz）、現在位置ステータス（SP）およびポーズナンバー（PN）が記憶される。

【0037】

前回の座標（Px, Py, Pz）は、プレイヤーキャラクターやNPCが壁等に衝突した場合に前回の座標に戻すために記憶される。現在位置ステータスデータ（PS）は、プレイヤーキャラクターが存在する座標の地形に関するデータであり、このデータに基づいて、加速度センサ出力値変換テーブル（空中、床面、氷面、水面）が選択される。ポーズナンバー（PN）は、キャラクターの状態（ポーズ）に関するデータ（例えば亀の表向きと裏向き等）である。

【0038】

図18は、表示用RAM25のメモリマップである。表示用RAM25には、CPU21がゲームプログラムを実行して得た表示用のデータが一時的に記憶される。表示用RAM25は、オブジェクトデータ記憶領域25a、スクロールカウンタデータ記憶領域25bおよびマップデータ記憶領域25cを備える。オブ

ジェクトデータ記憶領域 2 5 a には、ゲームに登場する全キャラクタのうち L C D 1 2 の表示領域内に存在するキャラクタについてのデータが記憶される。具体的には、X 座標、Y 座標、キャラクタ I D およびポーズナンバーが記憶される。

【 0 0 3 9 】

スクロールカウンタデータ記憶領域 2 5 b には、ゲームマップのうち L C D 1 2 に表示される領域の左上隅の座標が記憶される。マップデータ記憶領域 2 5 c には、ゲームマップのうち L C D 1 2 に表示される領域内のゲームマップデータが記憶される。

【 0 0 4 0 】

図 1 9 は、バックアップ R A M 3 5 のメモリマップである。バックアップ R A M 3 5 には、図 2 8 を参照して後述する 0 G 設定処理において設定される 0 G ポジションデータが記憶される。この 0 G ポジションデータは X Y 軸加速度センサ 3 1 が誤差を有するためにゲーム装置を水平に保ってもセンサ出力値が 0 にならないことに対処するためのものであり、ゲーム装置を水平に保ったときのセンサ出力値が 0 G ポジションデータとしてバックアップ R A M 3 5 に記憶され、ゲーム処理において、センサ出力値から差し引かれる。

【 0 0 4 1 】

図 2 0 から図 2 6 は、プログラム R O M 3 4 の加速度センサ出力値変換テーブル記憶領域 3 4 c に記憶される変換テーブルの詳細を示した図である。このテーブルには、X Y 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 のセンサ出力値 (I N x , I N y , I N z) および衝撃入力フラグ (F S) をゲーム処理に利用する際の利用方法や最大値を制限する等の補正処理のためのデータが記憶される。具体的には利用方法、補正比率、特殊補正条件および特殊補正数のデータが記憶される。このテーブルは複数記憶されており、ゲームマップ選択処理用テーブル、プレイヤーキャラクタ移動用テーブルおよび N P C 移動用テーブルが含まれる。

【 0 0 4 2 】

図 2 0 のゲームマップ選択処理用テーブルは、図 3 0 を参照して後述するゲームマップ選択処理において参照される。このテーブルによって、X Y 軸加速度セ

ンサの出力値 (IN_x , IN_y) はカメラ座標 (C_x , C_y) の変化量の計算をするために利用される。なお、補正比率は 2 倍なので、XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) の 2 倍だけカメラ座標 (C_x , C_y) は座標移動する。Z 軸接点スイッチ出力値 (IN_z) はマップ決定処理のために利用される。衝撃入力フラグ (FS) は利用しない。

【0043】

図 2 1 から図 2 4 のプレイヤーキャラクタ移動用テーブルは、図 3 3 を参照して後述するプレイヤーキャラクタ移動処理のステップ 3 3 における傾き移動処理とステップ 3 4 における衝撃移動処理において参照される。プレイヤーキャラクタ移動用テーブルは、空中用、床面用、氷面用および水中用のテーブルを含む。プレイヤーキャラクタが存在する座標の地形（現在位置ステータス）に応じてこれら複数の変換テーブルのうちいずれかが選択されて参照される。

【0044】

プレイヤーキャラクタ移動用テーブルにおいて、XY 軸加速度センサ 31 の出力値 X (IN_x) は、プレイヤーキャラクタの X 移動加速度の変化量 (dA_x) の計算のために利用され、出力値 Y (IN_y) は、 Y 移動加速度の変化量 (dA_y) の計算のために利用される。現在位置ステータスが「空中」の場合は、図 2 1 を参照して移動加速度の変化量 (dA_x , dA_y) はゼロである。「床面」の場合は、図 2 2 を参照すると補正比率が 2 倍であるから、XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) の 2 倍が移動加速度の変化量 (dA_x , dA_y) となる。また、特殊補正条件 1 により XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) が 20 より大きい場合には移動加速度の変化量 (dA_x , dA_y) は 40 に制限される。「氷面」の場合は、図 2 3 を参照して XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) の 3 倍が移動加速度の変化量 (dA_x , dA_y) となる（「氷面」では移動量大きい）。また、特殊補正条件 1 により XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) が 20 より大きい場合には移動加速度の変化量 (dA_x , dA_y) は 60 に制限される。「水中」の場合は、図 2 4 を参照して XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) の $1/2$ 倍が移動加速度の変化量 (dA_x , dA_y) となる（「水中」では移動量小さい）。また、特

殊補正条件 1 により X Y 軸加速度センサ 3 1 の出力値 (IN_x , IN_y) が 1 0 より大きい場合には移動加速度の変化量 (dA_x , dA_y) は 5 に制限される。

【0 0 4 5】

プレイヤーキャラクタ移動用テーブルにおいて、Z 軸接点スイッチ 3 2 の出力値 (IN_z) は、プレイヤーキャラクタの Z 移動加速度の変化量 (dA_z) の計算のために利用される。特殊補正条件はない。

【0 0 4 6】

プレイヤーキャラクタ移動用テーブルにおいて、衝撃入力フラグ (FS) は、X 移動加速度と Y 移動加速度の変化量 (dA_x , dA_y) に影響を与える。現在位置ステータスが「空中」および「水中」の場合は図 2 1 および図 2 4 を参照して衝撃入力フラグ (FS) は無視される。現在位置ステータスが「床面」の場合は、図 2 2 を参照して、X 移動加速度および Y 移動加速度の変化量 (dA_x , dA_y) をそれぞれ 3 倍する処理がされる。現在位置ステータスが「氷面」の場合は、図 2 3 を参照して、X 移動加速度および Y 移動加速度の変化量 (dA_x , dA_y) をそれぞれ 5 倍する処理がされる。このようにして、衝撃入力があった場合、「床面」と「氷面」では X 移動加速度および Y 移動加速度の変化量 (dA_x , dA_y) が通常に比べて増加される (高速移動する)。

【0 0 4 7】

図 2 5 および図 2 6 の NPC 移動用テーブルは、図 3 4 を参照して後述する NPC 移動処理のステップ 4 4 における傾き移動処理とステップ 4 5 における衝撃移動処理において参照される。NPC 移動用テーブルは、表向き用と裏向き用のテーブルを含む。NPC である亀のポーズ (表向きか裏向きか) に応じてこれら 2 つの変換テーブルのうちいずれかが選択されて参照される。

【0 0 4 8】

NPC 移動用テーブルにおいて、X Y 軸加速度センサ 3 1 の出力値 X (IN_x) は、NPC の X 移動加速度の変化量 (dA_x) の計算のために利用され、出力値 Y (IN_y) は、Y 移動加速度の変化量 (dA_y) の計算のために利用される。「表向き」の場合は、図 2 5 を参照すると補正比率が $1/2$ 倍であるから、X Y 軸加速度センサ 3 1 の出力値 (IN_x , IN_y) の $1/2$ 倍が X 移動加速度お

よびY移動加速度の変化量 (dA_x , dA_y) となる。また、特殊補正条件1によりXY軸加速度センサ31の出力値 (IN_x , IN_y) が10より小さい場合には移動加速度の変化量 (dA_x , dA_y) は0になる(「表向き」の場合は少しの傾き入力では亀は踏ん張るので滑らない)。また、特殊補正条件2によりXY軸加速度センサ31の出力値 (IN_x , IN_y) が20より大きい場合には、移動加速度の変化量 (dA_x , dA_y) は10に制限される。「裏向き」の場合は、図26を参照してXY軸加速度センサ31の出力値 (IN_x , IN_y) の2倍がX移動加速度およびY移動加速度の変化量 (dA_x , dA_y) となる(「表向き」に比べて「裏向き」の方が亀は滑りやすいので移動量が多い)。また、特殊補正条件1によりXY軸加速度センサ31の出力値 (IN_x , IN_y) が20より大きい場合には、移動加速度の変化量 (dA_x , dA_y) は40に制限される。

【0049】

NPC移動用テーブルにおいて、Z軸接点スイッチの出力値 (IN_z) は亀の表裏の逆転判定に利用される。Z軸接点スイッチ出力値が1になる毎に亀は表裏の状態を繰り返す。衝撃入力フラグ (FS) はNPC移動処理には利用しない。

【0050】

図27はメインルーチンのフローチャートである。ゲーム装置本体10にカートリッジ30を差して、ゲーム装置本体10の電源をONにすると、図27に示すメインルーチンが開始される。まずステップ11において、初回起動時であるかまたはプレイヤーが0G設定リクエスト(例えば、図1の操作スイッチ13bを押しながら起動)があったか否かが判断される。初回起動時でなく0G設定リクエストもない場合はステップ13に進む。初回起動時かまたは0G設定リクエストがあった場合は、ステップ12において、図28を参照して後述する0G設定処理が行われた後、ステップ14に進む。ステップ14において、図29を参照して後述するニュートラルポジション設定処理が行われた後、ステップ17に進む。ここで、ニュートラルポジション設定とは、ゲームプレイする際のゲーム装置の基準の傾きを設定することであり、お勧めポジション設定とは、ゲーム内容に応じて適切なニュートラルポジションに関するデータ(プログラムROM34

のお勧めポジション照準目標座標 3 4 d) を予めゲームプログラムに記憶させておいて、そのデータに基づいてニュートラルポジションを設定することである。

【 0 0 5 1 】

ステップ 1 7 において、図 3 0 を参照して後述するゲームマップ選択処理が行われ、複数のゲームマップのうちのいずれかがプレイヤによって選択される。ステップ 1 7 の後、メインループに進む。

【 0 0 5 2 】

ステップ 1 9 からステップ 2 9 までがメインループであり、ゲームオーバーになるかまたはゲームクリアになるまで繰り返し処理される。ステップ 1 9 において、ワーク RAM 2 6 のキャラクターデータ 2 6 g の座標 (X, Y, Z) およびポーズナンバー (PN)、プログラム ROM 3 4 のオブジェクトキャラクターデータ 3 4 a およびマップデータ 3 4 b に基づいて表示用 RAM 2 5 に必要なデータが書込まれ、表示用 RAM 2 5 に記憶されたデータに基づいて LCD 1 2 にゲーム画面が表示される。ステップ 2 0 において、図 3 1 を参照して後述するセンサ出力読取処理が行われて、XY 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 の出力値がセンサインターフェース 3 3 を介して読取られ、補正される。ステップ 2 0 の後、ステップ 2 1 においてニュートラルポジション設定要求があったか否かが判断され、要求がない場合はステップ 2 3 に進み、要求があった場合はステップ 2 2 に進んでニュートラルポジション設定処理が行われ、ニュートラルポジションの再設定がされた後、ステップ 1 9 に戻る。これは、一つの操作スイッチ（例えば図 1 における操作スイッチ 1 3 e）をニュートラルポジション設定専用の操作スイッチに割り当てておいて、この操作スイッチ 1 3 e を押すことによってゲーム中においても、常にニュートラルポジションの再設定を可能にすることを意味する。

【 0 0 5 3 】

ステップ 2 3 において、衝撃入力フラグが ON であるか否かが判断される。衝撃入力フラグが OFF のばあいにはステップ 2 6 に進む。衝撃入力フラグが ON の場合はステップ 2 4 に進み、プレイヤキャラクターの現在座標の地形が水中であるか否かが判断される（現在位置ステータス (PS) に基づいて判断される）。水

中でないと判断されると、ステップ 2 6 に進む。水中であると判断されるとステップ 2 5 において波の発生処理が行われる（前述の 1 5 中段に示すような画面表示となる）。具体的には、センサ出力値 X (IN_x) を X 軸方向のベクトル成分としセンサ出力値 Y (IN_y) を Y 軸方向のベクトル成分として合成したベクトルの方向に、合成ベクトルの大きさに応じた波を発生する処理をする。プレイヤーはあたかも自分がゲーム装置に与えた衝撃がそのままゲーム空間中の環境（水）に反映されたかのような感覚をプレイヤーに与えることができる。ステップ 2 5 の後、ステップ 2 6 に進む。

【 0 0 5 4 】

ステップ 2 6 において、図 3 2 から図 3 6 を参照して後述する各オブジェクト移動処理が行われ、プレイヤーキャラクタおよび NPC の移動処理が行われる。ステップ 2 6 の後、ステップ 2 7 において、図 3 7 を参照して後述する衝突処理が行われ、プレイヤーキャラクタと NPC 等との衝突処理が行われる。ステップ 2 7 の後、ステップ 2 9 において図 4 0 を参照して後述する画面スクロール処理が行われる。

【 0 0 5 5 】

図 2 8 は 0 G 設定処理のフローチャートである。このサブルーチンでは、ゲーム装置（具体的には LCD 1 2 の表示面）を水平に把持したときの XY 軸加速度センサ 3 1 の出力値を 0 G ポジションデータとしてバックアップ RAM 3 5 に記憶しておく処理を行う。

【 0 0 5 6 】

まずステップ 1 2 1 において、「地面と水平にあわせたら操作スイッチを押してください」という表示が LCD 1 2 にされて、ゲーム装置（具体的には LCD 1 2 の表示面）が水平になるようにプレイヤーに要求する。ステップ 1 2 2 において操作スイッチの入力処理が行われ、ステップ 1 2 3 において、決定のための操作スイッチ（例えば図 1 における操作スイッチ 1 3 b）が押されたことが判断されると、ステップ 1 2 4 において Z 軸接点スイッチが ON か否かが判断される。 Z 軸接点スイッチが ON の場合は、ステップ 1 2 5 において警告音を発生してステップ 1 2 1 に戻る。 Z 軸接点スイッチが ON になっている場合は、LCD 1 2

の表示面が下方を向いた状態になっているので、プレイヤーに再度設定することを要求するものである。ステップ 1 2 4 において、Z 軸接点スイッチが OFF であると判断された場合は、ステップ 1 2 6 において、このときの X Y 軸加速度センサ 3 1 の出力値がバックアップ RAM 3 5 に 0 G ポジションデータとして記憶される。

【 0 0 5 7 】

図 2 9 は、ニュートラルポジション設定処理のフローチャートである。このサブルーチンでは、ゲームプレイしやすいゲーム装置の把持角度をプレイヤーが任意に決定して、そのときの X Y 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 の出力値をニュートラルポジションデータとしてワーク RAM 2 6 に記憶しておく処理を行う。

【 0 0 5 8 】

まずステップ 1 4 1 において、「遊びやすい角度にあわせたら操作スイッチを押してください」という表示が LCD 1 2 にされる。ステップ 1 4 2 において、操作スイッチ入力処理が行われ、ステップ 1 4 3 において、決定のための操作スイッチ（例えば図 1 の操作スイッチ 1 3 b）が押されたことが判断されると、ステップ 1 4 4 において、このときの X Y 軸加速度センサ 3 1 の出力値から前述の 0 G ポジションデータを引く補正がされた後（ニュートラルポジションデータは水平状態からの傾きに対応したデータとする）、ステップ 1 4 5 において、X Y 軸加速度センサの出力補正值（ステップ 1 4 4 の計算結果）と Z 軸接点スイッチ 3 2 の出力値がワーク RAM 2 6 のニュートラルポジションデータ記憶領域 2 6 a にニュートラルポジションデータとして記憶される。

【 0 0 5 9 】

図 3 0 はゲームマップ選択処理のフローチャートである。このサブルーチンでは、ゲームプログラムに記憶されている複数のゲームマップのうちからプレイヤーがいずれか一つを選択する。ゲームマップ選択処理の画面は例えば前述の図 1 3 のように表示される。LCD 1 2 にはゲームマップ選択用マップの一部領域が表示される。プレイヤーは、X 軸方向または Y 軸方向にスライド入力することによって表示領域を移動させて、マップアイコン（図 1 3 の A, B, C, D）を表示領

域内に表示されるようにしてからZ軸方向に運動入力する。Z軸方向に運動入力（または衝撃入力）したときに表示領域内に表示されているコースアイコンに対応したゲームコースが選択されたことになる。

【0060】

まずステップ171において、カメラ座標（C_x，C_y）が初期化される。その後、ステップ172において、カメラ座標（C_x，C_y）に基づいてゲームマップ選択用マップの一部領域がLCD12に表示される。ステップ173において、図31を参照して後述するセンサ出力読取処理が行われて、XY軸加速度センサ31およびZ軸接点スイッチ32の出力値が読取られ、補正される。ステップ174において、図20に示すゲームマップ選択処理用テーブルが参照されて、センサ出力値（I_{Nx}，I_{Ny}）に基づいてカメラ座標（C_x，C_y）が変化される。具体的には、補正比率が2倍であるから、センサー出力値（I_{Nx}，I_{Ny}）の2倍の値だけカメラ座標（C_x，C_y）を変化させる。例えばセンサー出力値（I_{Nx}）の値が5であるときは、カメラ座標（C_x）が+10される。ステップ175において、カメラ座標（C_x，C_y）に基づく表示領域がゲームマップ選択用マップの範囲外であるか否かが判断され、範囲外でない場合はステップ177に進み、範囲外である場合は、ステップ176において、ゲームマップ選択用マップの端領域が表示されるように補正された後、ステップ177に進む。ステップ177において、Z軸接点スイッチ32がONであるか否かが判断される。Z軸接点スイッチ32がOFFであると判断されると、ステップ172に戻る。Z軸接点スイッチ32がONであると判断されると、ステップ178において、マップアイコン（図13のA，B，C，D）のうちのいずれか一つがLCD12の表示領域内に表示されているか否かが判断される。マップアイコンが表示領域内に表示されていないことが判断されると、ステップ179において、警告音を発生してステップ172に戻る。マップアイコンが表示領域内に表示されていることが判断されると、ステップ181において、表示されているマップアイコンに対応したゲームマップナンバー（MN）がワークRAM26に記憶される。

【0061】

図 3 1 はセンサ出力読取処理のフローチャートである。このサブルーチンでは、X Y 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 の出力値が読取られ、補正される。具体的には、センサインターフェース 3 3 のラッチ 3 3 4 およびラッチ 3 3 5 のデータから加速度センサ出力値 (I N x, I N y) および Z 軸接点スイッチ出力値 (I N z) が読み出される。さらに、0 G ポジションデータおよびニュートラルポジションデータで補正する処理がされる。

【0 0 6 2】

ステップ 2 0 1 において、センサインターフェース 3 3 のラッチ 3 3 4 およびラッチ 3 3 5 のデータが読み込まれる。ステップ 2 0 2 において、ラッチデータから加速度センサ出力値 (I N x, I N y) および Z 軸接点スイッチ出力値 (I N z) が読み出されてワーク RAM 2 6 の加速度センサ出力値記憶領域 2 6 b に記憶される。ステップ 2 0 3 において、衝撃入力があったか否かが判断される。具体的には、加速度センサ出力値 X (I N x) を X 軸方向のベクトル成分とし、加速度センサ出力値 Y (I N y) を Y 軸方向のベクトル成分として合成したベクトルの大きさが一定値以上か否かが判断される。一定値以上であると判断されると、ステップ 2 0 4 において、衝撃入力フラグ (F S) が O N に設定された後、ステップ 2 0 6 に進む。合成ベクトルの大きさが一定値より小さいと判断されると、ステップ 2 0 5 において、衝撃入力フラグ (F S) が O F F に設定された後、ステップ 2 0 6 に進む。ステップ 2 0 6 において、ステップ 2 0 2 において記憶された加速度センサ出力値記憶領域 2 6 b のデータからバックアップ RAM 3 5 に記憶された 0 G ポジションデータを引く処理がされる。ステップ 2 0 6 の後、ステップ 2 0 7 において、さらにニュートラルポジションデータで補正した値が、加速度センサ出力記憶領域 2 6 b に I N x, I N y および I N z として再記憶される。

【0 0 6 3】

ニュートラルポジションデータによる補正は、具体的には、加速度センサ出力値 X (I N x) および加速度センサ出力値 Y (I N y) については、ニュートラルポジションデータ (N P x, N P y) の値を引く処理がされる。Z 軸接点スイッチ出力値 (I N z) については、ニュートラルポジションデータ (N P z) の

値が1の場合に、0と1を反転させる処理がされる。

【0064】

図32から図36はオブジェクト移動処理のフローチャートである。図32は、オブジェクト移動処理のメインルーチンのフローチャートである。ステップ261において、図33を参照して後述するプレイヤーキャラクタ移動処理が行われる。ステップ262において、図34を参照して後述するNPC移動処理が行われる。このNPC移動処理はNPCの数だけ繰り返される。

【0065】

図33は、プレイヤーキャラクタ移動処理のフローチャートである。ステップ31において、プレイヤーキャラクタの現在の座標(X, Y, Z)が、前回の座標(Px, Py, Pz)としてコピーされて記憶される。これは、図37を参照して後述する衝突処理において、プレイヤーキャラクタが壁に衝突したときに前回の座標に戻すために必要である。ステップ32において、移動加速度の変化量(dAx, dAy, dAz)が初期化された後、ステップ33において、傾き移動処理が行われる。傾き移動処理では、前述の図21から図24に示す変換テーブルのうちプレイヤーキャラクタの現在位置ステータスに応じて適切なものを参照して、プレイヤーキャラクタのX移動加速度とY移動加速度の変化量を算出する処理が行われる。この処理によってゲーム装置の傾き(傾き入力)に応じてキャラクタが転がる(滑る)ように移動加速度の変化量(dAx, dAy)が決定される。さらに、ステップ34において、衝撃移動処理が行われる。衝撃移動処理では、前述の図21から図24に示す変換テーブルのうち適切なものを参照して、プレイヤーキャラクタのX移動加速度とY移動加速度の変化量を増加させる処理が行われる。この処理によって、衝撃入力がされたときに、プレイヤーキャラクタがダッシュ(高速移動)するように移動加速度の変化量(dAx, dAy)を増加する処理がされる。ステップ35において、図35を参照して後述するジャンプ移動処理が行われる。ステップ35の後、ステップ36において、前述の図27のフローチャートにおけるステップ25において波発生処理がされたか否かが判断される。波が発生していないことが判断されると、ステップ38に進む。波が発生したことが判断されると、ステップ37において、図36を参照して後述する波移

動処理が行われた後、ステップ 3 8 に進む。ステップ 3 8 において、ステップ 3 3 からステップ 3 7 までの傾き移動処理、衝撃移動処理、ジャンプ処理、波移動処理で計算された移動加速度の変化量 (dA_x , dA_y , dA_z) に基づいて移動加速度 (A_x , A_y , A_z) が計算され、移動加速度 (A_x , A_y , A_z) に基づいて速度 (V_x , V_y , V_z) が計算される。ステップ 3 9 において、速度 (V_x , V_y , V_z) に基づいて座標 (X , Y , Z) が計算される。

【0066】

図 3 4 は NPC 移動処理のフローチャートである。ステップ 4 1 において、現在の座標 (X , Y , Z) が前回の座標 (P_x , P_y , P_z) にコピーして記憶される。ステップ 4 2 において、移動加速度の変化量 (dA_x , dA_y , dA_z) が初期化される。ステップ 4 3 において、ゲームプログラムに基づく NPC の自律移動処理が行われる。具体的には例えば亀は乱数値に基づいて移動加速度の変化量 (dA_x , dA_y , dA_z) が決定される。ステップ 4 3 の後、ステップ 4 4 において、傾き移動処理が行われる。傾き移動処理では、前述の図 2 5 または図 2 6 に示す変換テーブルのうち NPC のポーズナンバーに応じて適切なものを参照して、NPC の X 移動加速度と Y 移動加速度の変化量を算出する処理が行われる。さらに、ステップ 4 5 において、衝撃移動処理が行われるが、本実施例の場合は、NPC は衝撃入力による影響を受けない。ステップ 4 5 の後、ステップ 4 6 において、前述の図 2 7 のフローチャートにおけるステップ 2 5 において波発生処理がされたか否かが判断される。波が発生していないことが判断されると、ステップ 4 8 に進む。波が発生したことが判断されると、ステップ 4 7 において、図 3 6 を参照して後述する波移動処理が行われた後、ステップ 4 8 に進む。ステップ 4 8 において、ステップ 4 3 からステップ 4 7 までの自律移動処理、傾き移動処理、衝撃移動処理、波移動処理で計算された移動加速度の変化量 (dA_x , dA_y , dA_z) に基づいて移動加速度 (A_x , A_y , A_z) が計算され、移動加速度 (A_x , A_y , A_z) に基づいて速度 (V_x , V_y , V_z) が計算される。ステップ 4 9 において、速度 (V_x , V_y , V_z) に基づいて座標 (X , Y , Z) が計算される。ステップ 5 1 において、Z 軸接点スイッチ出力値 (IN_z) が 1 か否かが判断される。Z 軸接点スイッチ出力値 (IN_z) が 0 の場合は N

PC移動処理サブルーチンを終了する。Z軸接点スイッチ出力値（IN_z）が1の場合は、ステップ52において表向きと裏向きの反転処理をする。具体的には、ワークRAM26のキャラクタデータのポーズナンバー（PN）を変化させる。

【0067】

図35はジャンプ処理のフローチャートである。このサブルーチンでは、Z軸方向の運動入力があったとき、プレイヤーキャラクタをジャンプさせる処理がされ、Z軸方向の運動入力がなくてかつプレイヤーキャラクタが「空中」にいるときに降下させる処理がされる。

【0068】

ステップ351において、Z軸接点スイッチ出力値（IN_z）が1か否かが判断される。Z軸接点スイッチ出力値（IN_z）が1の場合には、ステップ352において、現在位置ステータス（PS）が「空中」に設定された後、ステップ353において、Z移動加速度の変化量（dA_z）が1にされる。ステップ351において、Z軸接点スイッチ出力値（IN_z）が0であると判断されると、ステップ354において、プレイヤーキャラクタが「空中」にいるか否かが判断され、「空中」にいない場合はジャンプ処理を終了する。ステップ354において、空中にいる場合は、ステップ355において、Z移動加速度の変化量（dA_z）が-1にされた後、ジャンプ処理を終了する。

【0069】

図36は波移動処理のフローチャートである。このサブルーチンでは、プレイヤーの衝撃入力により発生した波によるプレイヤーキャラクタやNPCの移動加速度の変化量を計算する処理がされる。ステップ361において、現在位置ステータスが読み込まれ、ステップ362において、波の影響を受ける位置か否か（すなわち「水中」か否か）が判断される。波の影響を受けない位置であると判断されると、波移動処理を終了する。波の影響を受ける位置であると判断されると、ステップ363において、波の影響によるX移動加速度の変化量とY移動加速度の変化量が計算されて、傾き移動処理および衝撃移動処理において計算されたX移動加速度の変化量とY移動加速度の変化量に加算される。

【 0 0 7 0 】

図 3 7 は衝突処理のフローチャートである。ステップ 2 7 1 からステップ 2 7 5 において NPC 衝突判定処理が行われる。この NPC 衝突判定処理は NPC の数だけ繰り返される。ステップ 2 7 1 において、NPC が壁と衝突したか否かが判断される。衝突したと判断されるとステップ 2 7 3 に進む。壁と衝突していないと判断されるとステップ 2 7 2 に進み、別の NPC と衝突したか否かが判断される。別の NPC と衝突したと判断されるとステップ 2 7 3 に進む。別の NPC と衝突していないと判断されるとステップ 2 7 5 に進む。壁または別の NPC に衝突したと判定された場合は、ステップ 2 7 3 において、衝突音が発生された後、ステップ 2 7 4 において、NPC の座標 (X, Y, Z) を前回の座標 (P x, P y, P z) に戻す処理が行われた後、ステップ 2 7 5 に進む。

【 0 0 7 1 】

ステップ 2 7 5 において、NPC の現在位置ステータスが検出されてワーク RAM 2 6 に記憶される。ステップ 2 7 5 の後、ステップ 2 7 6 において、プレイヤーキャラクタが壁と衝突したか否かが判断される。壁に衝突していないと判断された場合には、ステップ 2 7 9 に進む。壁に衝突したと判断された場合には、ステップ 2 7 7 において、衝突音が発生された後、ステップ 2 7 8 において、プレイヤーキャラクタの座標 (X, Y, Z) を前回の座標 (P x, P y, P z) に戻す処理がされた後、ステップ 2 7 9 に進む。

ステップ 2 7 9 において、プレイヤーキャラクタの現在位置ステータスが検出されてワーク RAM 2 6 に記憶される。ステップ 2 7 9 の後、ステップ 2 8 1 において、プレイヤーキャラクタが NPC と衝突したか否かが判断される。NPC と衝突したと判断されると、ステップ 2 8 2 において、NPC を消滅する処理がされる。ステップ 2 8 2 の後、ステップ 2 8 3 において、すべての NPC が消滅したか否かが判断される。すべての NPC が消滅したことが判断されるとステップ 2 8 4 においてゲームクリア処理が行われる。ステップ 2 8 1 において、NPC と衝突していないと判断された場合およびステップ 2 8 3 においてすべての NPC が消滅していないと判断された場合にはステップ 2 8 5 に進む。ステップ 2 8 5 において、プレイヤーキャラクタが穴に落ちたか否かが判断される。穴に落ちたと判

断された場合は、ステップ 2 8 6 においてゲームオーバー処理が行われる。穴に落ちていないと判断された場合は、衝突処理を終了する。

【 0 0 7 2 】

図 3 8 および図 3 9 は画面スクロールを示す画面の一例である。画面にはプレイヤーキャラクタであるボール 6 1, N P C である亀 6 2 a ~ 6 2 c, 迷路を構成する壁 6 3 および穴 6 4 が表示されている。点線 6 5 は画面スクロールの限界を示すものである（点線 6 5 は実際には L C D 1 2 には表示されない）。前述のようにゲームマップは L C D 1 2 の表示領域より大きな仮想マップであり、L C D 1 2 にはゲームマップのうちプレイヤーキャラクタ 6 1 の周辺の一部領域が表示される。プレイヤーがゲーム装置を傾ける等してプレイヤーキャラクタ 6 1 が点線 6 5 より外側領域に移動しようとする、画面をスクロールして L C D 1 2 に表示されるゲームマップ表示領域を移動させ、さらにプレイヤーキャラクタ 6 1 および N P C 6 2 をスクロールした分だけ画面の中央方向に移動表示する。この画面スクロールによって、より広いゲームマップ上でのゲームを楽しむことができる。

【 0 0 7 3 】

例えば、図 3 8 に示すようにプレイヤーキャラクタが点線 6 5 を超えて左側の領域に移動しようとする、ゲームマップの表示領域を左にスクロールさせ、プレイヤーキャラクタ 6 1 および N P C 6 2 をスクロールさせた分だけ右に移動表示させる（図 3 9）。なお、傾き入力の数値に応じてスクロールする速度を変化させても良い。

【 0 0 7 4 】

図 4 0 は、画面スクロール処理のフローチャートである。ステップ 2 9 1 において、プレイヤーキャラクタがスクロールエリアを X 軸のマイナス方向に外れたか否かが判断される。ここで、スクロールエリアとは、図 3 8 における点線 6 5 で囲まれる領域のことである。X 軸のマイナス方向に外れていないことが判断されるとステップ 2 9 4 に進む。X 軸のマイナス方向に外れたことが判断されると、ステップ 2 9 2 において、L C D 1 2 に現在表示されている領域がゲームマップの左端領域か否かが判断される。左端領域であると判断された場合は、ステップ 2 9 4 に進む。左端領域でないと判断された場合は、ステップ 2 9 3 において、表

示用 R A M 2 5 に記憶されたスクロールカウンタ X 座標 (S C x) を一定量減らす処理がされた後、ステップ 2 9 4 に進む。ステップ 2 9 4 において、プレイヤーキャラクタがスクロールエリアを X 軸のプラス方向に外れたか否かが判断される。X 軸のプラス方向に外れていないことが判断されると、ステップ 2 9 7 に進む。X 軸のプラス方向に外れたことが判断されると、ステップ 2 9 5 において、L C D 1 2 に現在表示されている領域がゲームマップの右端領域か否かが判断される。右端領域であることが判断されるとステップ 2 9 7 に進む。右端領域でないと判断された場合は、ステップ 2 9 6 において、スクロールカウンタ X 座標 (S C x) を一定量増やす処理がされた後、ステップ 2 9 7 に進む。

【 0 0 7 5 】

ステップ 2 9 7 において、プレイヤーキャラクタがスクロールエリアを Y 軸のマイナス方向に外れたか否かが判断される。Y 軸のマイナス方向に外れていないことが判断されると、ステップ 3 0 1 に進む。Y 軸のマイナス方向に外れたことが判断されると、ステップ 2 9 8 において、L C D 1 2 に現在表示されている領域がゲームマップの上端領域か否かが判断される。上端領域であることが判断されるとステップ 3 0 1 に進む。上端領域でないと判断された場合は、ステップ 2 9 9 において、スクロールカウンタ Y 座標 (S C y) を一定量減らす処理がされた後、ステップ 3 0 1 に進む。ステップ 3 0 1 において、プレイヤーキャラクタがスクロールエリアを Y 軸のプラス方向に外れたか否かが判断される。Y 軸のプラス方向に外れていないことが判断されると、画面スクロール処理を終了する。Y 軸のプラス方向に外れたことが判断されると、ステップ 3 0 2 において、L C D 1 2 に現在表示されている領域がゲームマップの下端領域か否かが判断される。下端領域であることが判断されると、画面スクロール処理を終了する。下端領域でないと判断された場合は、ステップ 3 0 3 において、スクロールカウンタ Y 座標 (S C y) を一定量増やす処理がされた後、画面スクロール処理を終了する。

【 0 0 7 6 】

(第 2 実施例)

次に、図 4 1 ～ 図 4 9 を参照して、本発明の第 2 の実施例のゲーム装置を説明する。第 2 の実施例のゲーム装置の外観図、X Y Z 軸定義図、ブロック図、セン

サインターフェースの計測原理図、Z軸接点スイッチの構造図は、第1の実施例における図1～図7と共通であり、説明を省略する。

【0077】

図41は、本実施例のゲーム画面の一例を示した図である。このゲームは、プレイヤーがゲーム装置に衝撃を与えることによってゲーム空間の地形を隆起させることにより、ゲームキャラクタの移動を制御して楽しむゲームである。

【0078】

図41(a)に示すように、ゲームキャラクタである亀81と地形隆起キャラクタ82がゲーム画面に表示される。図41(b)に示すように、亀81はゲームプログラムによって自立的に移動する。図41(b)に示す状態において、ゲーム装置にZ軸方向の衝撃入力を与えると、図41(c)に示すように、地形隆起キャラクタ82は隆起して高く大きく表示され、それによって亀81は滑る移動制御をされる（前進していた亀82が地形隆起によって後退する）。このような処理をすることによって、ゲーム装置にZ軸方向の衝撃を与えたときにゲーム空間である地形がエネルギーを与えられて地形が隆起するかなのような感覚をプレイヤーに与えることができる。

【0079】

図42はZ軸方向の衝撃入力による地形隆起処理を示すゲーム画面の一例である。図42(a)において、外枠12'はゲーム空間全体を示しており、内枠12はLCD12に表示される表示領域を示している。ゲーム空間はLCD12の表示領域より大きな世界であり、LCD12にはゲーム空間の一部領域が表示される。ゲーム空間には地形隆起キャラクタ82が12個存在し（82a～82l）、亀キャラクタ81が3匹存在する（81a～81c）。このうち、LCD12には4個の地形隆起キャラクタ（82a, 82b, 82e, 82f）と1匹の亀キャラクタ（82a）が表示されている。

【0080】

図42(a)に示す状態において、ゲーム装置にZ軸方向の衝撃入力を与えると、図42(b)に示すように、12個の地形隆起キャラクタ（82a～82l, ゲーム空間全体の地形隆起キャラクタ）は1段階隆起して高く大きく表示され

る。このとき、地形が隆起した地点に存在する亀キャラクタ（81 a および 81 b）は地形の隆起によって滑って移動する表示がされる。

【0081】

図42（b）に示す状態において、Aボタン（操作スイッチ13 b）を操作しつつZ軸方向の衝撃入力を与えると、LCD12に表示されている4個の地形隆起キャラクタ（82 a, 82 b, 82 e, 82 f）のみさらに1段階隆起して高く大きく表示される。このときも同様に、地形が隆起した地点に存在する亀キャラクタ（81 a）は地形の隆起によって滑って移動する表示がされる。このように処理することによって、Aボタンを押しつつZ軸方向の衝撃入力を与えた場合には、LCD12に表示された領域に限定したゲーム空間に衝撃によるエネルギーが与えられたかのような感覚をプレイヤに与えることができる。

【0082】

なお、図示しないが、図42（b）に示す状態において、Bボタン（操作スイッチ13 c）を操作しつつZ軸方向の衝撃入力を与えると、LCD12に表示されていない8個の地形隆起キャラクタ（82 c, 82 d, 82 g, 82 h, 82 i ~ 82 l）のみ1段階隆起して高く大きく表示される。このときも同様に、地形が隆起した地点に存在する亀キャラクタ（81 b, 81 c）は地形の隆起によって滑って移動する表示がされる。このように処理することによって、Bボタンを押しつつZ軸方向の衝撃入力を与えた場合には、LCD12に表示されていない領域に限定してゲーム空間に衝撃によるエネルギーが与えられたかのような感覚をプレイヤに与えることができる。

【0083】

図43は、ゲーム空間の表示領域のスクロール処理を示すゲーム画面の一例である。ゲーム装置をスライド入力（第1の実施例における図9を参照）をすることによってゲーム空間の表示領域がスクロールする。例えば、図43（a）では、LCD12に地形隆起キャラクタ82 a, 82 b, 82 e, 82 f および亀キャラクタ81 aが表示されている。この状態において、ゲーム装置をY軸のマイナス方向にスライドさせると、ゲーム空間の表示領域は下方にスクロールして、図43（b）に示すように、地形キャラクタ82 e, 82 f および亀キャラクタ

8 1 a が表示されることになる。

【 0 0 8 4 】

また、図 4 3 (b) に示す状態において、ゲーム装置を X 軸のプラス方向にスライドさせると、ゲーム空間の表示領域は右方にスクロールして、図 4 3 (c) に示すように、地形キャラクタ 8 2 f および亀キャラクタ 8 1 a が表示されることになる。このような処理をすることによって、プレイヤは L C D 1 2 より大きなゲーム空間でゲームを楽しむことができる。また、前述のように、A ボタンや B ボタンによって表示領域の内外に限定してゲーム空間に影響を与える（地形を隆起させる）ことができるので、複雑なゲームを楽しむことができる。

【 0 0 8 5 】

図 4 4 は、X Y 軸方向の衝撃入力による温度上昇画面制御を示す図である。亀キャラクタ 8 1 a ～ 8 1 c は、前述のようにゲームプログラムによって自立的な移動をするが、この自律移動は、温度が上昇することによって活発になる（具体的には移動量が増加する）。図 4 4 (a) に示す状態において、X Y 軸方向の衝撃入力（第 1 の実施例における図 1 1 を参照）をすると、温度パラメータが上昇し、亀キャラクタ 8 1 a ～ 8 1 c が活発に移動する表示がされる。このような処理をすることによって、ゲーム装置に X Y 軸方向の衝撃を与えたときにゲーム空間にエネルギーが与えられて温度が上昇するかのような感覚をプレイヤーに与えることができる。

【 0 0 8 6 】

以下、図 4 5 および図 4 6 を参照して、メモリに記憶されるデータを説明する。

図 4 5 は、プログラム ROM 3 4 のメモリマップである。プログラム ROM 3 4 には、C P U 2 1 によって実行されるゲームプログラムおよびゲームデータが記憶される。プログラム ROM 3 4 は、具体的には、オブジェクトキャラクタデータ記憶領域 3 4 2 a，マップデータ記憶領域 3 4 2 b，地形隆起ポイントデータ記憶領域 3 4 2 c，スクロールの限界値データ記憶領域 3 4 2 d，加速度センサ出力値変換テーブル記憶領域 3 4 2 e およびゲームプログラム記憶領域 3 4 2 f を含む。オブジェクトキャラクタデータ記憶領域 3 4 2 a およびマップデータ

記憶領域 3 4 2 b には、オブジェクトキャラクタおよびゲームマップのグラフィックデータが記憶される。地形隆起ポイントデータ記憶領域 3 4 2 c には、前述の図 4 2 に示されるような地形隆起キャラクタ（8 2 a ～ 8 2 l）のそれぞれについてゲーム空間における位置データ（X 座標および Y 座標； $P x 1 \sim P x 1 2$ ， $P y 1 \sim P y 1 2$ ）が記憶される。スクロールの限界値データ記憶領域 3 4 2 d には、ゲーム空間をスクロール表示する際に、ゲーム空間の上下左右端になった場合にスクロールをしないようにするためにスクロールの限界値を示すデータ（ $SC x m a x$ ， $SC y m a x$ ）が記憶される。

【 0 0 8 7 】

加速度センサ出力値変換テーブル記憶領域 3 4 2 d には、X Y 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 の出力値を変換してゲームプログラムで利用するための変換テーブルが記憶されている。具体的には、前述の第 1 実施例の変換テーブル（図 2 0 から図 2 6）と同様のデータが記憶されており、センサ出力値 X（ $I N x$ ）およびセンサ出力値 Y（ $I N y$ ）が、後述の図 4 8 を参照して後述する視界移動処理において、スクロールカウンタ X 座標（ $SC x$ ）および Y 座標（ $SC y$ ）の変化量の計算に利用されることが定義されている。これによってゲーム装置をスライド入力（第 1 の実施例における図 9 を参照）をすることによってゲーム空間の表示領域がスクロールし、視界が移動する処理がされることになる。また、Z 軸接点スイッチ出力値（ $I N z$ ）は地形の隆起判定に利用されることが定義され、衝撃入力フラグ（ $F S$ ）は温度上昇の判定に利用されることが定義されている。

【 0 0 8 8 】

ゲームプログラム記憶領域 3 4 2 f には、CPU 2 1 によって実行されるゲームプログラムが記憶される。具体的には、図 4 7 を参照して後述するメインプログラム、第 1 の実施例における図 3 1 と同様のセンサ出力読取プログラム、図 4 8 を参照して後述する視界移動プログラム、図 4 9 を参照して後述する地形隆起プログラム、温度上昇プログラム、亀キャラクタ制御プログラムやその他のプログラムが記憶される。

【0089】

図46はワークRAM26のメモリマップである。ワークRAM26には、CPU21がゲームプログラムを実行する際の一時的なデータが記憶される。具体的には、加速度センサ出力値記憶領域262a、衝撃入力フラグ記憶領域262b、地形隆起データ記憶領域262c、温度データ記憶領域262dおよびキャラクターデータ記憶領域262eが含まれる。

【0090】

加速度センサ出力値記憶領域262aおよび衝撃入力フラグ記憶領域262bに記憶されるデータは第1の実施例と同様であるので説明を省略する。地形隆起データ記憶領域262cには、それぞれの地形隆起ポイントについての高さデータが記憶される。高さデータは、図49を参照して後述する地形隆起処理においてZ軸方向の衝撃入力に応じて変更される。このデータに基づいてそれぞれの地形隆起ポイントについて地形隆起キャラクタの表示状態が決定される。例えば、高さデータが1の場合には図42(a)の82aのように表示され、高さデータが2の場合には図42(b)の82aのように表示され、高さデータが3の場合には図42(c)の82aのように地形隆起キャラクタが表示される。

【0091】

温度データ記憶領域には、ゲーム空間の温度データが記憶される。温度データは、温度上昇処理(図47に示すメインプログラムのステップ64)においてXY軸方向の衝撃入力に応じて変更される。このデータは亀キャラクタ制御処理(自律移動、図47に示すメインプログラムのステップ65)に影響を与える。

キャラクターデータ記憶領域262eには、亀キャラクタの数だけ、座標データ(X、Y、Z)および前回の座標データ(Px、Py、Pz)が記憶される。

表示用RAMのメモリマップは第1の実施例における図18と同様であるので説明を省略する。

【0092】

以下、図47～図49を参照して、ゲームプログラムの処理の流れを説明する。

図47はメインルーチンのフローチャートである。ゲーム装置本体10にカー

トリッジ 30 を差して、ゲーム装置本体 10 の電源を ON にすると、図 47 に示すメインルーチンが開始される。第 2 の実施例においても、第 1 の実施例と同じように、0 G ポジション設定処理やニュートラルポジションの設定処理をおこなっても良いが、説明を簡単にするため省略することにする。

【0093】

まずステップ 61 において、第 1 の実施例における図 31 と同様のセンサ出力読取処理が行われて、XY 軸加速度センサ 31 および Z 軸接点スイッチ 32 の出力値がセンサインターフェース 33 を介して読取られる（0 G ポジションデータおよびニュートラルポジションデータによる補正は省略）。ステップ 61 の後、ステップ 62 において、図 48 を参照して後述する視界移動処理（ゲーム空間の表示領域のスクロール処理）が行われる。ステップ 62 の後、ステップ 63 において、図 49 を参照して後述する地形隆起処理が行われる。ステップ 63 の後、ステップ 64 において、温度上昇処理が行われる。温度上昇処理では、まず XY 軸方向の衝撃入力があったか否かが判定され、XY 軸方向の衝撃入力があった場合には、温度パラメータ（T）を 1 増加する処理がされる。ステップ 64 の後、ステップ 65 において、亀キャラクタ制御処理が行われる。亀キャラクタ制御処理では、まず自律移動による亀キャラクタの移動処理が行われる。具体的には、例えば、乱数値によって亀キャラクタの移動量を計算する処理が行われる。なお、温度（T）が高いときには亀キャラクタの自立移動量が大きくなるように制御される。この後、地形隆起による亀キャラクタの移動処理が行われる。具体的には、亀キャラクタの下に地形が隆起した場合に、亀キャラクタを滑らせて移動させる処理がされる。なお、亀キャラクタ制御処理は亀キャラクタの数だけ繰り返される。

【0094】

ステップ 65 の後、ステップ 66 において、前述の視界移動処理、地形隆起処理、亀キャラクタ制御処理の処理結果に基づいて、ゲーム空間のスクロール表示および地形隆起オブジェクトや亀キャラクタの表示処理がされる。なお、地形隆起処理によって地形隆起ポイントの高さが増加された場合には、地形隆起キャラクタを高く大きく表示するとともに、地形が隆起していることを連想させるよう

な音を発生するとより効果的である。ステップ 6 6 の後、ステップ 6 7 において、ゲームオーバーか否かが判断される。例えば、所定時間を過ぎた場合にゲームオーバーにする等ゲーム内容に応じた適当な条件によってゲームオーバーの判定がされる。ステップ 6 7 において、ゲームオーバーであると判定されるとメインルーチンを終了する。ステップ 6 7 において、ゲームオーバーでないと判定されると、ステップ 6 1 に戻る。

【 0 0 9 5 】

図 4 8 は視界移動処理のフローチャートである。まず、ステップ 6 2 1 において、変換テーブルを参照して、スクロールカウンタ X 座標 (S C x) および Y 座標 (S C y) の変更処理が行われる。ステップ 6 2 1 の後、ステップ 6 2 2 ～ステップ 6 2 9 において、ゲーム空間の端を超えてスクロールしようとしているか否かが判断されて、ゲーム空間の端を超えてスクロールしようとしている場合には、スクロールカウンタの値 (S C x 、 S C y) を適正な値にする処理が行われる。

【 0 0 9 6 】

ステップ 6 2 2 において、スクロールカウンタ X 座標 (S C x) がスクロール限界値 X 座標 (S C x m a x) を超えているか否かが判断され、超えていないと判断された場合には、ステップ 6 2 4 に進む。ステップ 6 2 2 において、超えていると判断された場合には、ステップ 6 2 3 に進み、スクロールカウンタ X 座標 (S C x) の値がスクロール限界値 X 座標 (S C x m a x) に設定された後、ステップ 6 2 4 に進む。

【 0 0 9 7 】

ステップ 6 2 4 において、スクロールカウンタ X 座標 (S C x) が 0 より小さいか否かが判断され、0 以上であると判断された場合は、ステップ 6 2 6 に進む。ステップ 6 2 4 において、0 より小さいと判断された場合は、ステップ 6 2 5 に進み、スクロールカウンタ X 座標 (S C x) の値が 0 に設定された後、ステップ 6 2 6 に進む。

【 0 0 9 8 】

ステップ 6 2 6 において、スクロールカウンタ Y 座標 (S C y) がスクロール

限界値Y座標（SCymax）を超えているか否かが判断され、超えていないと判断された場合は、ステップ628に進む。ステップ626において、超えていると判断された場合は、ステップ627に進み、スクロールカウンタY座標（SCy）の値がスクロール限界値Y座標（SCymax）に設定された後、ステップ628に進む。

【0099】

ステップ628において、スクロールカウンタY座標（SCy）が0より小さいか否かが判断され、0以上であると判断された場合は、視界移動処理を終了する。ステップ628において、0より小さいと判断された場合は、ステップ629に進み、スクロールカウンタY座標（SCy）の値が0に設定された後、視界移動処理を終了する。

【0100】

図49は地形隆起処理のフローチャートである。まず、ステップ631において、Z軸接点スイッチの出力があったか否か（すなわち、Z軸方向の衝撃入力があったか否か）が判断され、Z軸接点スイッチの出力がなかったと判断された場合は、地形隆起処理を終了する。Z軸接点スイッチの出力があったと判断された場合は、ステップ632に進む。ステップ632において、Aボタン（操作スイッチ13b）が押されているか否かが判断され、Aボタンが押されていると判断された場合は、ステップ633に進み、LCDに表示されている領域内の地形隆起ポイントの高さ（H）をそれぞれ1増加させる処理がされる。ステップ633の後、地形隆起処理を終了する。

【0101】

ステップ632において、Aボタンが押されていないことが判断されると、ステップ634に進み、Bボタン（操作スイッチ13c）が押されているか否かが判断される。Bボタンが押されていることが判断されると、ステップ635に進み、LCDに表示されている領域外の地形隆起ポイントの高さ（H）をそれぞれ1増加させる処理がされる。ステップ635の後、地形隆起処理を終了する。ステップ634において、Bボタンが押されていないことが判断されると、ステップ636において、全地形隆起ポイントの高さ（H）をそれぞれ1増加させる処

理がされた後、地形隆起処理を終了する。

【 0 1 0 2 】

(第 3 実施例)

次に、図 5 0 ～図 5 9 を参照して、本発明の第 3 の実施例を説明する。このゲームは、ゲーム装置をフライパンや包丁であるかのように動かして仮想的な料理を楽しむゲームである。

【 0 1 0 3 】

図 5 0 ～図 5 3 は、ゲーム画面の一例である。図 5 0 において、ゲーム画面には、プレイヤーキャラクタ 9 1 とキッチン 9 2 とコンロ 9 3 とフライパン 9 4 と机 9 5 とまな板 9 6 が表示される。A ボタン（操作スイッチ 1 3 b）を押すと、図 5 1 および図 5 2 を参照して後述するフライパン空間処理が開始する。また、B ボタン（操作スイッチ 1 3 c）を押すと、図 5 3 を参照して後述する包丁空間処理が開始する。

【 0 1 0 4 】

図 5 1 および図 5 2 は、フライパン空間処理のゲーム画面の一例である。フライパン空間処理では、ゲーム装置をフライパンのように操作して、目玉焼きを調理するゲームをする。図 5 1 (a) において、ゲーム画面には、フライパン 9 4 と卵 9 7 が表示されている。図 5 1 (a) に示す状態において、ゲーム装置を Y 軸を中心にマイナス方向に傾けると、図 5 1 (b) に示すように、卵 9 7 はフライパンの左方へ移動表示される。また、図 5 1 (b) に示す状態において、ゲーム装置を X 軸を中心としてプラス方向に傾けると、卵 9 7 はフライパンの下方へ移動表示される。このように処理することによって、ゲーム装置をあたかもフライパンのように操作して、卵がフライパンの傾きによって移動しているかのような感覚をプレイヤーに与えることができる。

【 0 1 0 5 】

図 5 2 (a) に示す状態において、ゲーム装置の Z 軸方向に衝撃入力を与えると、図 5 2 (b) に示すように、卵 9 7 はフライパン 9 4 から離れて上方にジャンプしているような表示がされ、その後、図 5 2 (c) または (d) に示すように、卵 9 7 は着地する表示がされる。このとき、図 5 2 (a) に示すように、Z

軸方向に衝撃入力をした時の卵 9 7 の位置がフライパン 9 4 の端の方である場合には、卵 9 7 はフライパン 9 4 からみだしてジャンプし着地する（図 5 2（c））こととなり、失敗となる。なお、図 5 2（b）に示す状態において、ゲーム装置をスライド移動させることによって、卵 9 7 とフライパン 9 4 との相対的な位置関係を修正して、卵 9 7 をフライパン 9 4 の中に着地させることが可能である（図 5 2（d））。このように処理することによって、ゲーム装置をあたかもフライパンのように操作して、ジャンプした卵をフライパンで受け止めるかのような感覚をプレイヤに与えることができる。

【0106】

図 5 3 は包丁空間処理のゲーム画面の一例である。包丁空間処理では、ゲーム装置を包丁のように操作して、キャベツを千切りするゲームをする。図 5 3（a）において、ゲーム画面には、包丁 9 8 とキャベツ 9 9 が表示されている。図 5 3（a）に示す状態において、ゲーム装置を X 軸のプラス方向にスライド移動させると、図 5 3（b）に示すように、包丁 9 8 に対してキャベツ 9 9 が左に移動する表示がされる（包丁 9 8 は常にゲーム画面の中央に表示されるため、キャベツ 9 9 が相対的に左に移動する表示がされる）。このように処理することによって、ゲーム装置をあたかも包丁のように操作して、キャベツと包丁との位置関係を調節してキャベツを切断する位置を調節しているかのような感覚をプレイヤに与えることができる。

【0107】

さらに、図 5 3（b）に示す状態において、ゲーム装置を上下に運動（Z 軸方向の運動入力）させると、キャベツ 9 9 が包丁 9 8 によって千切りされる表示がされる。この際、キャベツが切断される音を発生させるとより効果的である。

【0108】

以下、図 5 4 を参照して、メモリに記憶されるデータを説明する。なお、プログラム ROM 3 4 には、第 1 実施例のプログラム ROM（図 1 6）と略同様のプログラムが記憶されるが、加速度センサ出力値変換テーブル記憶領域には、フライパン用テーブルと卵ジャンプ時用テーブルと包丁用テーブルが記憶され、ゲームプログラム記憶領域には、メインプログラム、センサ出力読み取りプログラム

、フライパン空間プログラム、卵ジャンププログラム、包丁空間プログラムおよびその他のプログラムが記憶される。なお、加速度センサ出力値変換テーブルのフライパン用テーブルは、図 5 6 を参照して後述するフライパン空間プログラムで参照され、卵ジャンプ時用テーブルは、図 5 8 を参照して後述する卵ジャンププログラムで参照され、包丁用テーブルは、図 5 7 を参照して後述する包丁空間プログラムで参照される。

【0109】

フライパン用テーブルには、XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) は卵の X および Y 座標 (E_x , E_y) の変化量の計算に利用されることが定義される。これによって、ゲーム装置を傾き入力 (第 1 の実施例における図 10 を参照) したときに卵の表示位置が変更され、フライパンの上を卵が滑るように表示制御される。また、座標 Z 軸接点スイッチ 32 の出力値 (IN_z) は、卵のジャンプ判定に利用され、衝撃入力フラグ (FS) は利用されないことが定義される。

【0110】

卵ジャンプ時用テーブルには、XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) は卵の X および Y 座標 (E_x , E_y) の変化量の計算に利用されることが定義される。これによって、卵がジャンプしている間にゲーム装置をスライド入力 (第 1 の実施例における図 9 を参照) したときに卵の表示位置が変更され、フライパンと卵の相対位置が変更されるように表示制御される。なお、卵ジャンプ時用テーブルにおいて補正比率はマイナス値が定義される。なぜなら、本実施例では、フライパンはゲーム画面に固定的に表示され、卵がフライパンに対して相対的に移動するように表示されるため、ゲーム装置のスライド移動の方向とは逆の方向に卵を移動表示する必要があるためである。また、Z 軸接点スイッチ 32 の出力値 (IN_z) および衝撃入力フラグ (FS) は利用されない。

【0111】

包丁用テーブルには、XY 軸加速度センサ 31 の出力値 (IN_x , IN_y) はキャベツの X および Y 座標 (CA_x , CA_y) の変化量の計算に利用されることが定義されている。これによって、ゲーム装置をスライド入力したときにキャベ

ツの表示位置が変更され、キャベツと包丁の相対位置が変更されるように表示制御される。なお、包丁用テーブルは卵ジャンプ時用テーブルと同様に補正比率はマイナス値が定義される。なぜなら、本実施例では、包丁はゲーム画面に固定的に表示され、キャベツが包丁に対して相対的に移動するように表示されるため、ゲーム装置のスライド移動の方向とは逆の方向にキャベツを移動表示する必要があるためである。また、Z軸接点スイッチ32の出力値(INz)は、キャベツの切断処理の判定に利用され、衝撃入力フラグ(FS)は利用されないことが定義される。

【0112】

図54はワークRAM26のメモリマップである。ワークRAM26には、CPU21がゲームプログラムを実行する際の一時的なデータが記憶される。具体的には、加速度センサ出力値記憶領域263a、衝撃入力フラグ記憶領域263b、卵データ記憶領域263cおよびキャベツデータ記憶領域263dが含まれる。

【0113】

加速度センサ出力値記憶領域263aおよび衝撃入力フラグ記憶領域263bに記憶されるデータは第1の実施例と同様であるので説明を省略する。

卵データ記憶領域263cには、卵のX座標(Ex)、Y座標(Ey)、高さ(Eh)および焼け具合(Ef)のデータが記憶される。キャベツデータ記憶領域263dには、キャベツのX座標(CAx)、Y座標(CAy)および切断割合(CAc)のデータが記憶される。

表示用RAMのメモリマップは第1の実施例における図18と同様であるので説明を省略する。

【0114】

以下、図55～図59を参照して、ゲームプログラムの処理の流れを説明する。

図55はメインルーチンのフローチャートである。ゲーム装置本体10にカートリッジ30を差して、ゲーム装置本体10の電源をONにすると、図55に示すメインルーチンが開始される。第3の実施例においても、第1の実施例と同じ

ように、0 G ポジション設定処理やニュートラルポジションの設定処理をおこなっても良いが、説明を簡単にするため省略することにする。

【0 1 1 5】

まずステップ7 1において、第1の実施例における図3 1と同様のセンサ出力読取処理が行われて、X Y 軸加速度センサ3 1およびZ 軸接点スイッチ3 2の出力値がセンサインターフェース3 3を介して読取られる（0 G ポジションデータおよびニュートラルポジションデータによる補正は省略）。ステップ7 1の後、ステップ7 2において、A ボタン（操作スイッチ1 3 b）が押されているか否かが判断される。ステップ7 2において、A ボタンが押されていることが判断されると、ステップ7 3に進み、図5 7を参照して後述する包丁空間処理が行われた後、ステップ7 6に進む。

【0 1 1 6】

ステップ7 2において、A ボタンが押されていないことが判断されると、ステップ7 4に進み、B ボタン（操作スイッチ1 3 c）が押されているか否かが判断される。ステップ7 4において、B ボタンが押されていないことが判断されると、ステップ7 6に進む。ステップ7 4において、B ボタンが押されていることが判断されると、ステップ7 5に進み、図5 6を参照して後述するフライパン空間処理が行われた後、ステップ7 6に進む。

【0 1 1 7】

ステップ7 6において、ゲームオーバーか否かが判断される。具体的には、例えば、所定時間を過ぎた場合にゲームオーバーにする等ゲーム内容に応じた適当な条件によってゲームオーバーの判定がされる。ステップ7 6において、ゲームオーバーでないと判断されると、ステップ7 1に戻る。ステップ7 6において、ゲームオーバーであると判断されると、メインルーチンを終了する。

【0 1 1 8】

図5 6はフライパン空間処理のフローチャートである。まず、ステップ7 7 1において、フライパン用テーブルを参照して、卵X座標（E x）および卵Y座標（E y）の変更処理が行われる。ステップ7 7 1の後、ステップ7 7 2において、図5 8を参照して後述する卵ジャンプ処理が行われる。ステップ7 7 2の後、

ステップ 7 7 3 において、卵焼け具合 (E f) を 1 増加させる処理がおこなわれる。ステップ 7 7 3 の後、ステップ 7 7 4 において、卵焼け具合 (E f) が 1 0 0 以上になったか否かが判断される。卵焼け具合 (E f) が 1 0 0 より小さいことが判断されると、フライパン空間処理を終了する。卵焼け具合 (E f) が 1 0 0 以上になったことが判断されると、ステップ 7 7 5 に進み、卵成功処理が行なわれる。卵成功処理では、例えば、卵の調理が完成した画面が表示され、得点が加算される処理が行われる。ステップ 7 7 5 の後、フライパン空間処理を終了する。

【 0 1 1 9 】

図 5 7 は包丁空間処理のフローチャートである。まず、ステップ 7 4 1 において、包丁用テーブルを参照して、キャベツ X 座標 (C A x) およびキャベツ Y 座標 (C A y) の変更処理が行われる。ステップ 7 4 1 の後、ステップ 7 4 2 において、図 5 9 を参照して後述するキャベツ切断処理が行われる。ステップ 7 4 2 の後、ステップ 7 4 3 において、キャベツ切断割合 (C A c) が 1 0 0 以上になったか否かが判断される。キャベツ切断割合 (C A c) が 1 0 0 より小さいことが判断されると、包丁空間処理を終了する。キャベツ切断割合 (C A c) が 1 0 0 以上になったことが判断されると、ステップ 7 4 4 に進み、キャベツ成功処理が行なわれる。キャベツ成功処理では、例えば、キャベツの切断が完成した画面が表示され、得点が加算される処理が行われる。ステップ 7 4 4 の後、包丁空間処理を終了する。

【 0 1 2 0 】

図 5 8 は卵ジャンプ処理のフローチャートである。まず、ステップ 7 7 2 a において、Z 軸接点スイッチの出力があったか否か（すなわち、Z 軸方向の衝撃入力があったか否か）が判断される。ステップ 7 7 2 a において、Z 軸接点スイッチの出力がなかったことが判断されると、卵ジャンプ処理を終了する。ステップ 7 7 2 a において、Z 軸接点スイッチの出力があったことが判断されると、ステップ 7 7 2 b において、卵がジャンプしている表示が行われる。ステップ 7 7 2 b の後、ステップ 7 7 2 c において、卵の高さ (E h) が C H (所定値) に設定される。ステップ 7 7 2 c の後、ステップ 7 7 2 d において、第 1 の実施例にお

ける図 3 1 と同様のセンサ出力読取処理が行われて、X Y 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 の出力値がセンサインターフェース 3 3 を介して読取られる（0 G ポジションデータおよびニュートラルポジションデータによる補正は省略）。ステップ 7 7 2 d の後、ステップ 7 7 2 e において、卵ジャンプ時用テーブルを参照して、卵 X 座標（E x）および卵 Y 座標（E y）の変更処理が行われる。ステップ 7 7 2 e の後、ステップ 7 7 2 f において、卵の高さ（E h）を 1 減少させる処理が行われる。ステップ 7 7 2 f の後、ステップ 7 7 2 g において、卵の X 座標（E x），Y 座標（E y），高さ（E h）に基づいて表示処理が行われる。ステップ 7 7 2 g の後、ステップ 7 7 2 h において、卵が着地したか否か、すなわち卵の高さ（E h）が 0 になったか否かが判断される。ステップ 7 7 2 h において、卵が着地していないことが判断されると、ステップ 7 7 2 d に戻る。ステップ 7 7 2 h において、卵が着地したことが判断されると、ステップ 7 7 2 i において、卵の着地位置がフライパン内か否かが判断され、フライパン内であると判断された場合にはステップ 7 7 2 j において、ジャンプ成功処理が行われた後、卵ジャンプ処理を終了する。ジャンプ成功処理では、例えば「成功」と表示しつつ成功の音楽を発生し、得点を加算する処理が行われる。ステップ 7 7 2 i において、卵の着地位置がフライパンの外であると判断された場合には、ステップ 7 7 2 k において、ジャンプ失敗処理が行われた後、卵ジャンプ処理を終了する。ジャンプ失敗処理では、例えば「失敗」と表示しつつ失敗の音楽を発生し、卵焼け具合（E f）を 0 にする処理が行われる（卵の調理をやり直させるようにする）。

【 0 1 2 1 】

図 5 9 はキャベツ切断処理のフローチャートである。まず、ステップ 7 4 2 a において、Z 軸接点スイッチの出力があったか否か（すなわち、Z 軸方向の運動入力があったか否か）が判断される。ステップ 7 4 2 a において、Z 軸接点スイッチの出力がなかったことが判断されると、キャベツ切断処理を終了する。ステップ 7 4 2 a において、Z 軸接点スイッチの出力があったことが判断されると、ステップ 7 4 2 b において、包丁の下にキャベツがあるか否かが判断される。ステップ 7 4 2 b において、包丁の下にキャベツがないことが判断されると、キャ

ベツ切断処理を終了する。ステップ742bにおいて、包丁の下にキャベツがあることが判断されると、ステップ742cにおいて、表示処理（キャベツが一定量切断される表示）がされる。ステップ742cの後、ステップ742dにおいて、キャベツの切断割合（CAc）を1増加する処理がされた後、キャベツ切断処理を終了する。

【0122】

（第4実施例）

次に、図60～図66を参照して、本発明の第4の実施例を説明する。図60はゲーム空間の概念図および複数のゲーム装置のゲーム画面の一例を示した図である。このゲームは、複数のゲーム装置で通信することによってゲーム空間を共有して、第1の実施例と同様のゲームを複数のプレイヤーでおこなって対戦（または協力）を楽しむゲームである。ゲーム空間である迷路板は複数のゲーム装置10および40に共通であり、ゲーム装置10のゲーム画像とゲーム装置40のゲーム画像は同じゲーム空間データに基づいている（ただし、それぞれのゲーム装置で視界が異なる）。第1のゲーム装置10のLCDには1点鎖線で示される範囲12が表示され、第2のゲーム装置40のLCDには点線で示される範囲42が表示される。第1の実施例と同様に、ゲーム装置の傾きに依じて、ゲーム空間である迷路板が傾いたことがシミュレートされるが、本実施例の場合は、ゲーム装置10の傾きとゲーム装置40の傾きを合成した値によって迷路板の傾きがシミュレートされる（一方のゲーム装置の傾きのみによって迷路版の傾きをシミュレートしてもよい）。ゲーム装置10のプレイヤーは自分のボール61aを操作するためにゲーム装置10を傾けて迷路板の傾きを操作しようとするが、一方でゲーム装置40のプレイヤーは自分のボール61bを操作するためにゲーム装置40を傾けて迷路板の傾きを操作しようとするので、お互いに自分の思う通りには迷路板の傾きを操作することができず、より複雑なゲームを楽しむことができる。なお、本実施例においては、通信ケーブル50を用いて2つのゲーム装置間の通信をおこなうこととするが、無線や携帯電話等の通信手段を利用しても良い。

【0123】

第4の実施例のプログラムROMには、第1の実施例のプログラムROM（図

16) と略同様のデータが記憶されるが、ゲームプログラム記憶領域には、第1の実施例の場合に追加して、図63および図64を参照して後述するマップ確認プログラムおよび図65および図66を参照して後述する通信割込プログラムがさらに記憶される。

【0124】

ゲームプログラム記憶領域に記憶されるプログラムのうち、メインプログラムとマップ確認プログラムと通信割込プログラムは、ゲーム装置10とゲーム装置40とで異なったプログラムとなっている。これは、ゲーム装置10を親機とし、ゲーム装置40を子機として通信処理を行うためであるが、詳細については、図61～図66を参照して後述する。

【0125】

第4の実施例のワークRAMには、第1の実施例のワークRAM17と略同様のデータが記憶されるが、第1の実施例の場合に追加して、合成データ記憶領域をさらに含む。合成データ記憶領域には、ゲーム装置10のXY軸加速度センサ31およびZ軸接点スイッチ32の出力値と、ゲーム装置40のXY軸加速度センサ31およびZ軸接点スイッチ32の出力値を合成した値がそれぞれ記憶される。

表示用RAMおよびバックアップRAMのメモリマップは第1の実施例における図18および図19と同様であるので説明を省略する。

【0126】

以下、図61～図66を参照して、ゲームプログラムの処理の流れを説明する。

図61はゲーム装置10で実行されるメインルーチンのフローチャートである。本実施例においては、説明を簡単にするために、OG設定処理、ニュートラルポジション設定処理および衝撃入力による波発生処理を省略しているが、第1の実施例と同様に、これらの処理を追加しても良い。

【0127】

まず、ステップ81pにおいて、第1の実施例における図30と同様のゲームマップ選択処理が行われる。ステップ81pの後、ステップ82pにおいて、図

6 3 を参照して後述する親機マップ確認処理が行われる。ステップ 8 2 p の後、ステップ 8 3 p に進む。

【 0 1 2 8 】

ステップ 8 3 p からステップ 8 5 p までがメインループであり、ゲームオーバーになるかまたはゲームクリアになるまで繰り返し処理される。ステップ 8 3 p において、ワーク RAM 2 6 のデータに基づいて表示用 RAM 2 5 に必要なデータが書込まれ、表示用 RAM 2 5 に記憶されたデータに基づいて LCD 1 2 にゲーム画面が表示される。ステップ 8 4 p において、第 1 の実施例における図 3 2 から図 3 6 と同様の各オブジェクト移動処理が行われ（波移動処理は省略）、プレイヤーキャラクタおよび NPC の移動処理が行われる。ステップ 8 4 p の後、ステップ 8 5 p において、第 1 の実施例における図 3 7 と同様の衝突処理が行われ、プレイヤーキャラクタと NPC 等との衝突処理が行われる。ステップ 8 5 p の後、ステップ 8 6 p において第 1 の実施例における図 4 0 と同様の画面スクロール処理が行われる。

【 0 1 2 9 】

図 6 2 はゲーム装置 4 0 で実行されるメインルーチンのフローチャートである。本実施例においては、説明を簡単にするために、OG 設定処理、ニュートラルポジション設定処理および衝撃入力による波発生処理を省略しているが、第 1 の実施例と同様にこれらの処理を追加しても良い。

【 0 1 3 0 】

まず、ステップ 8 1 c において、第 1 の実施例における図 3 0 と同様のゲームマップ選択処理が行われる。ステップ 8 1 c の後、ステップ 8 2 c において、図 6 4 を参照して後述する子機マップ確認処理が行われる。ステップ 8 2 c の後、ステップ 8 3 c に進む。

【 0 1 3 1 】

ステップ 8 3 c からステップ 8 8 c までがメインループであり、ゲームオーバーになるかまたはゲームクリアになるまで繰り返し処理される。まず、ステップ 8 3 c において、ワーク RAM 2 6 のデータに基づいて表示用 RAM 2 5 に必要なデータが書込まれ、表示用 RAM 2 5 に記憶されたデータに基づいて LCD 1

2にゲーム画面が表示される。ステップ83cの後、ステップ84cにおいて、第1の実施例における図31と同様のセンサ出力読取処理が行われて、XY軸加速度センサ31およびZ軸接点スイッチ32の出力値がセンサインターフェース33を介して読取られる（0Gポジションデータおよびニュートラルポジションデータによる補正は省略）。ステップ84cの後、ステップ85cにおいて、割込信号および先のステップ84cにおいて読込まれてワークRAM26に記憶された加速度センサ出力値データ（INx, INy, INz）がゲーム装置10に送信される。ゲーム装置10側では、この割込信号を受けて、図65を参照して後述する親機通信割込処理が開始される。ステップ85cの後、ステップ86cにおいて、第1の実施例における図32から図36と同様の各オブジェクト移動処理が行われ（波移動処理は省略）、プレイヤーキャラクタおよびNPCの移動処理が行われる。ステップ86cの後、ステップ87cにおいて、第1の実施例における図37と同様の衝突処理が行われ、プレイヤーキャラクタとNPC等との衝突処理が行われる。ステップ87cの後、ステップ88cにおいて第1の実施例における図40と同様の画面スクロール処理が行われる。

【0132】

図63は、ゲーム装置10で実行される親機マップ確認処理のフローチャートである。まず、ステップ87p1において、自身のワークRAM26に記憶されたマップナンバーデータがゲーム装置40に送信される。ステップ87p1の後、ステップ87p2において、データの受信がおこなわれる。具体的には、図64を参照して後述する子機マップ確認処理のステップ87c3においてゲーム装置40から送信されるマップナンバーデータを受信する。ステップ87p3において、データが受信されたことが判断されると、ステップ87p4において、自身のマップナンバーデータと先のステップ87p2において受信したゲーム装置40のマップナンバーデータが一致するか否かが判断される。ステップ87p4において、マップナンバーデータが一致することが判断されると、親機マップ確認処理を終了する。ステップ87p4において、マップナンバーデータが一致しないことが判断されると、図61のメインルーチンのステップ81pのゲームマップ選択処理に戻る。

【 0 1 3 3 】

図 6 4 は、ゲーム装置 4 0 で実行される子機マップ確認処理のフローチャートである。まず、ステップ 8 7 c 1 において、データの受信が行われる。具体的には、前述の図 6 3 の親機マップ確認処理のステップ 8 7 p 1 においてゲーム装置 1 0 から送信されるマップナンバーデータを受信する。ステップ 8 7 c 2 においてデータが受信したことが判断されると、ステップ 8 7 c 3 において、自身のワーク RAM 2 6 に記憶されたマップナンバーデータがゲーム装置 1 0 に送信される。ステップ 8 7 c 3 の後、ステップ 8 7 c 4 において、自身のマップナンバーデータと先のステップ 8 7 c 1 において受信したゲーム装置 1 0 のマップナンバーデータが一致するか否かが判断される。ステップ 8 7 c 4 において、マップナンバーデータが一致することが判断されると、子機マップ確認処理を終了する。ステップ 8 7 c 4 において、マップナンバーデータが一致しないことが判断されると、図 6 2 のメインルーチンのステップ 8 1 c のゲームマップ選択処理に戻る。

【 0 1 3 4 】

図 6 5 は、ゲーム装置 1 0 で実行される親機通信割込み処理のフローチャートである。この処理は、前述の図 6 2 に示すゲーム装置 4 0 のメインルーチンのステップ 8 5 c において送信される割込信号によって処理が開始される。まず、ステップ 9 1 p において、データの受信を行う。具体的には、前述の図 6 2 に示すゲーム装置 4 0 のメインルーチンのステップ 8 5 c において送信されるゲーム装置 4 0 の加速度センサ出力値を受信する。ステップ 9 1 p の後、ステップ 9 2 p において、第 1 の実施例における図 3 1 と同様のセンサ出力読取処理が行われて、X Y 軸加速度センサ 3 1 および Z 軸接点スイッチ 3 2 の出力値がセンサインターフェース 3 3 を介して読取られる（0 G ポジションデータおよびニュートラルポジションデータによる補正は省略）。ステップ 9 2 p の後、ステップ 9 3 p において、先のステップ 9 1 p で受信したゲーム装置 4 0 の加速度センサ出力値と先のステップ 9 2 p で読取ったゲーム装置 1 0 の加速度センサ出力値を合成する。ここで、合成とは、単に加算する計算処理をおこなっても良いし、例えば 2 つの値に重みをつけて加算する等、より複雑な計算式により 2 つの値から合成値を

算出しても良い。ステップ 9 3 p の後、ステップ 9 4 p において、割込信号および先のステップ 9 3 p で計算された合成データがゲーム装置 4 0 に送信される。

【0 1 3 5】

図 6 6 は、ゲーム装置 4 0 で実行される子機通信割込み処理のフローチャートである。この処理は、図 6 5 の親機通信割込み処理のステップ 9 4 p において送信される割込信号に応じて処理が開始される。ステップ 9 1 c においてゲーム装置 1 0 から合成データを受信して終了する。

【0 1 3 6】

上述の実施例は、携帯型ゲーム装置が検出手段を備えるものであったが、図 6 7 に示すように家庭用ゲーム機、パソコンまたは業務用ゲーム機等のコントローラが検出手段を備えるようにしてもよい。この場合には、プレイヤーはコントローラを傾けたり、運動や衝撃を与えることによってテレビジョン受信機等の表示装置に表示されるゲーム空間の制御をおこなう。例えば、図 6 8 に示すように、コントローラを傾けることによって表示装置に表示されたゲーム空間である板が傾く表示がされ、かつ、板の上のボールが転がることがシミュレートされる。コントローラを右に傾けると板が右に傾いてボールが右に転がり、コントローラを左に傾けると板が左に傾いてボールが左に転がることがシミュレートされる。

【0 1 3 7】

上述の実施例では、カートリッジに加速度センサを設けたが、携帯型ゲーム装置本体側に加速度センサを設けても良い。携帯型ゲーム装置本体側に加速度センサを設けた場合、カートリッジ毎に加速度センサを備える必要がなくコストを削減できる。また、ゲーム装置に用いる情報記憶媒体は、カートリッジに限るものではなく、PCカードのようなICカードであっても良い。

【0 1 3 8】

上述の第 1 の実施例では、ニュートラルポジションデータはワーク RAM 2 6 に記憶してゲームプレイ毎に設定するようにしたが、バックアップ RAM 3 5 に記憶して次のゲームプレイにおいても同じデータが利用可能にしても良い。

【0 1 3 9】

上述の第 1 の実施例では、ニュートラルポジションはプレイヤーが決定するよう

にしたが、予めゲームプログラムにニュートラルポジションデータを記憶しておいてこれを利用するようにしても良い。また、複数のニュートラルポジションデータを記憶しておいて、プレイヤーがいずれか一つを選択するようにしても良い。

【0140】

上述の第1の実施例では、ゲームキャラクタは、プレイヤーキャラクタ（ボール）と敵キャラクタ（亀）のみであったが、これらに加えて、プレイヤーキャラクタを助ける味方キャラクタや中立的なキャラクタ等のNPC（ノンプレイヤーキャラクタ）を登場させても良い。これらのNPCは、ゲームプログラムに基づいて自律移動されるが（自律移動しないNPCがあっても良い）、プレイヤーによる操作（傾き、運動または衝撃入力）に応じて移動や変形等させても良い。

【0141】

上述の第1の実施例では、ゲーム空間の制御は加速度センサの出力のみに基づいているが、操作スイッチに基づいてゲーム空間を制御する部分があってもよい。例えば、ピンボールゲームにおいて、ゲーム装置を傾けたり揺らしたりすることによってゲーム空間であるピンボール台を制御しつつ、操作スイッチを押した場合にフリッパーが動作するようなゲームが考えられる。また、いわゆる「落ちゲー」といわれる、落下する物体を積み上げて、積み上げ状態に応じて得点を計算するようなゲームにおいて、ゲーム装置を傾けたり揺らしたりすることによってゲーム空間を制御しつつ、操作スイッチで物体の向きを変えたり、衝撃入力で物体を高速移動させたり、Z軸方向の運動入力によって物体を変形させたりするようなゲームが考えられる。

【0142】

上述の第1の実施例では、ゲームキャラクタはゲーム装置の傾き（すなわち、ゲーム空間である迷路板の傾き）に応じて移動することとしたが、ゲーム装置の運動または衝撃に応じて移動するようにしてもよい。例えば、ゲーム装置をスライド移動させたときに、迷路板の壁が同じように移動したことがシミュレートされて、壁に接しているゲームキャラクタが壁に押されたかのように移動するような表示制御をおこなうことが考えられる。

【0143】

上述の第 1 の実施例では、プレイヤーキャラクタ（ボール）自体を移動表示したが、プレイヤーキャラクタは固定的に表示し、ゲーム空間をスクロール表示してプレイヤーキャラクタがゲーム空間を相対的に移動している表示処理としてもよい。

【0 1 4 4】

上述の第 4 の実施例は、2 人のプレイヤーが迷路板を傾けるという同一の制御を行うものであったが、2 人のプレイヤーが別々の制御を担当するようにしても良い。例えば、一人のプレイヤーはゲーム装置を傾けることによって迷路板を傾ける制御をし、他方のプレイヤーはゲーム装置に Z 軸方向の運動入力をしてゲームキャラクタをジャンプさせたり、X Y 軸方向の衝撃入力をして波を発生する制御をする等のゲームが考えられる。

【0 1 4 5】

上述の第 4 の実施例は、メインプログラム、マップ確認プログラムおよび通信割り込みプログラムについて、ゲーム装置 1 0 には親機用のプログラムが記憶され、ゲーム装置 4 0 には子機用のプログラムが記憶されていたが、ゲーム装置 1 0 およびゲーム装置 4 0 のそれぞれに親機用のプログラムと子機用のプログラムの両方を記憶しておいて、ゲーム開始に先立って、どちらを親機とし子機とするかを設定し、設定に従ってプログラムを選択するようにしてもよい。

【図面の簡単な説明】

【図 1】

本発明の一実施例の携帯型ゲーム装置の外観図である。

【図 2】

X Y Z 軸の定義を示した図である。

【図 3】

携帯型ゲーム装置のブロック図である。

【図 4】

センサインターフェースのブロック図である。

【図 5】

加速度センサの出力を計測する原理を示した図である。

【図 6】

Z 軸接点スイッチの構造を示した図である。

【図 7】

Z 軸接点スイッチが Z 軸方向の運動入力（または衝撃入力）を検出する場合の図である。

【図 8】

第 1 の実施例のゲーム画面の一例である。

【図 9】

スライド入力を示した図である。

【図 1 0】

傾き入力を示した図である。

【図 1 1】

X 軸方向または Y 軸方向の衝撃入力を示した図である。

【図 1 2】

Z 軸方向の運動入力（衝撃入力）を示した図である。

【図 1 3】

スライド入力の利用方法を示した図である。

【図 1 4】

傾き入力の利用方法を示した図である。

【図 1 5】

衝撃入力の利用方法を示した図である。

【図 1 6】

第 1 の実施例のプログラム ROM のメモリマップである。

【図 1 7】

第 1 の実施例のワーク RAM のメモリマップである。

【図 1 8】

第 1 の実施例の表示用 RAM のメモリマップである。

【図 1 9】

第 1 の実施例のバックアップ RAM のメモリマップである。

【図 2 0】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 1】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 2】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 3】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 4】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 5】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 6】

第 1 の実施例の加速度センサ出力変換テーブルである。

【図 2 7】

第 1 の実施例のメインルーチンのフローチャートである。

【図 2 8】

第 1 の実施例の 0 G 設定処理のフローチャートである。

【図 2 9】

第 1 の実施例のニュートラルポジション設定処理のフローチャートである。

【図 3 0】

第 1 の実施例のゲームマップ選択処理のフローチャートである。

【図 3 1】

第 1 の実施例のセンサ出力読取処理のフローチャートである。

【図 3 2】

第 1 の実施例の各オブジェクト移動処理のフローチャートである。

【図 3 3】

第 1 の実施例のプレイヤーキャラクタ移動処理のフローチャートである。

【図 3 4】

第 1 の実施例の NPC 移動処理のフローチャートである。

【図 3 5】

第 1 の実施例のジャンプ移動処理のフローチャートである。

【図 3 6】

第 1 の実施例の波移動処理のフローチャートである。

【図 3 7】

第 1 の実施例の衝突処理のフローチャートである。

【図 3 8】

第 1 の実施例の画面スクロールの説明図（スクロール前）である。

【図 3 9】

第 1 の実施例の画面スクロールの説明図（スクロール後）である。

【図 4 0】

第 1 の実施例の画面スクロール処理のフローチャートである。

【図 4 1】

第 2 の実施例のゲーム画面の一例である。

【図 4 2】

第 2 の実施例のゲーム画面（地形隆起処理）の一例である。

【図 4 3】

第 2 の実施例のゲーム画面（視界移動処理）の一例である。

【図 4 4】

第 2 の実施例のゲーム画面（温度上昇処理）の一例である。

【図 4 5】

第 2 の実施例のプログラム ROM のメモリマップである。

【図 4 6】

第 2 の実施例のワーク RAM のメモリマップである。

【図 4 7】

第 2 の実施例のメインルーチンのフローチャートである。

【図 4 8】

第 2 の実施例の視界移動処理のフローチャートである。

【図 4 9】

第 2 の実施例の地形隆起処理のフローチャートである。

【図 5 0】

第 3 の実施例のゲーム画面の一例である。

【図 5 1】

第 3 の実施例のゲーム画面（フライパン空間処理）の一例である。

【図 5 2】

第 3 の実施例のゲーム画面（フライパン空間処理）の一例である。

【図 5 3】

第 3 の実施例のゲーム画面（包丁空間処理）の一例である。

【図 5 4】

第 3 の実施例のワーク RAM のメモリマップである。

【図 5 5】

第 3 の実施例のメインルーチンのフローチャートである。

【図 5 6】

第 3 の実施例のフライパン空間処理のフローチャートである。

【図 5 7】

第 3 の実施例の包丁空間処理のフローチャートである。

【図 5 8】

第 3 の実施例の卵ジャンプ処理のフローチャートである。

【図 5 9】

第 3 の実施例のキャベツ切断処理のフローチャートである。

【図 6 0】

第 4 の実施例のゲーム画面の一例である。

【図 6 1】

第 4 の実施例のゲーム装置 1 0 のメインルーチンのフローチャートである。

【図 6 2】

第 4 の実施例のゲーム装置 4 0 のメインルーチンのフローチャートである。

【図 6 3】

第 4 の実施例の親機マップ確認処理のフローチャートである。

【図 6 4】

第 4 の実施例の子機マップ確認処理のフローチャートである。

【図 6 5】

第 4 の実施例の親機通信割込み処理のフローチャートである。

【図 6 6】

第 4 の実施例の子機通信割込み処理のフローチャートである。

【図 6 7】

本発明を家庭用ゲーム装置のコントローラに適用した場合の例である。

【図 6 8】

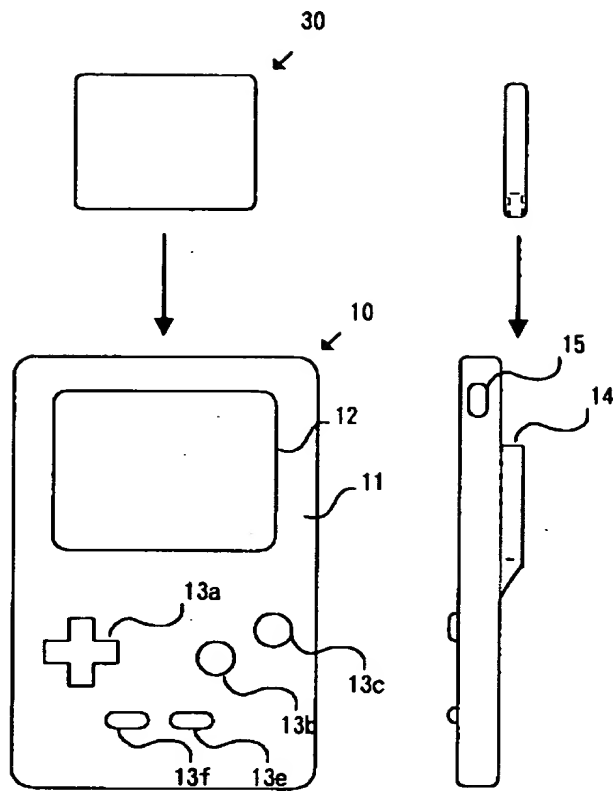
本発明を家庭用ゲーム装置のコントローラに適用した場合の画面例である。

【符号の説明】

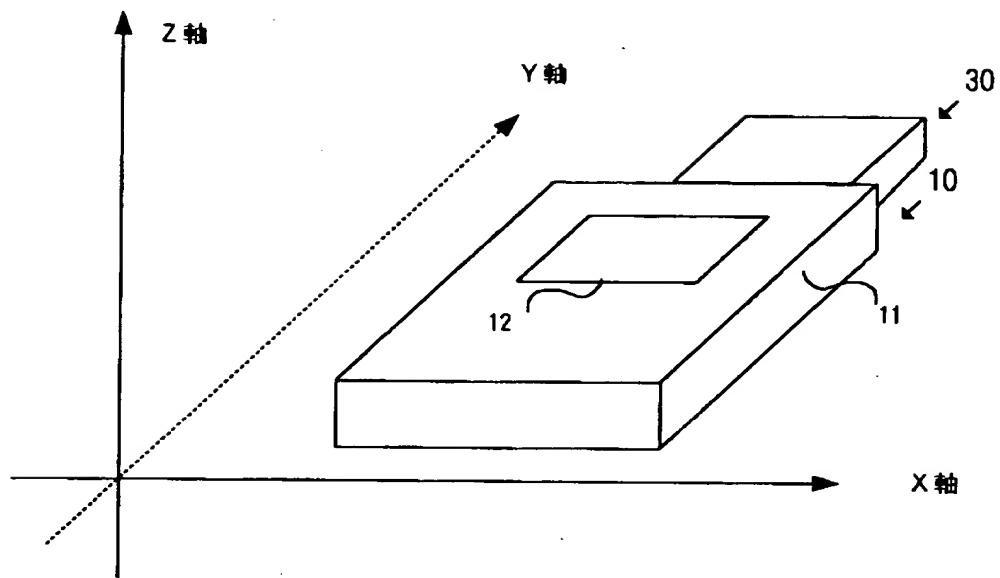
- 1 0 : ゲーム装置本体
- 1 2 : L C D
- 1 3 : 操作スイッチ
- 2 1 : C P U
- 2 5 : 表示用 R A M
- 2 6 : ワーク R A M
- 3 0 : ゲームカートリッジ
- 3 1 : X Y 軸加速度センサ
- 3 2 : Z 軸接点スイッチ
- 3 3 : センサインタフェース
- 3 4 : プログラム R O M
- 3 5 : バックアップ R A M

【書類名】 図面

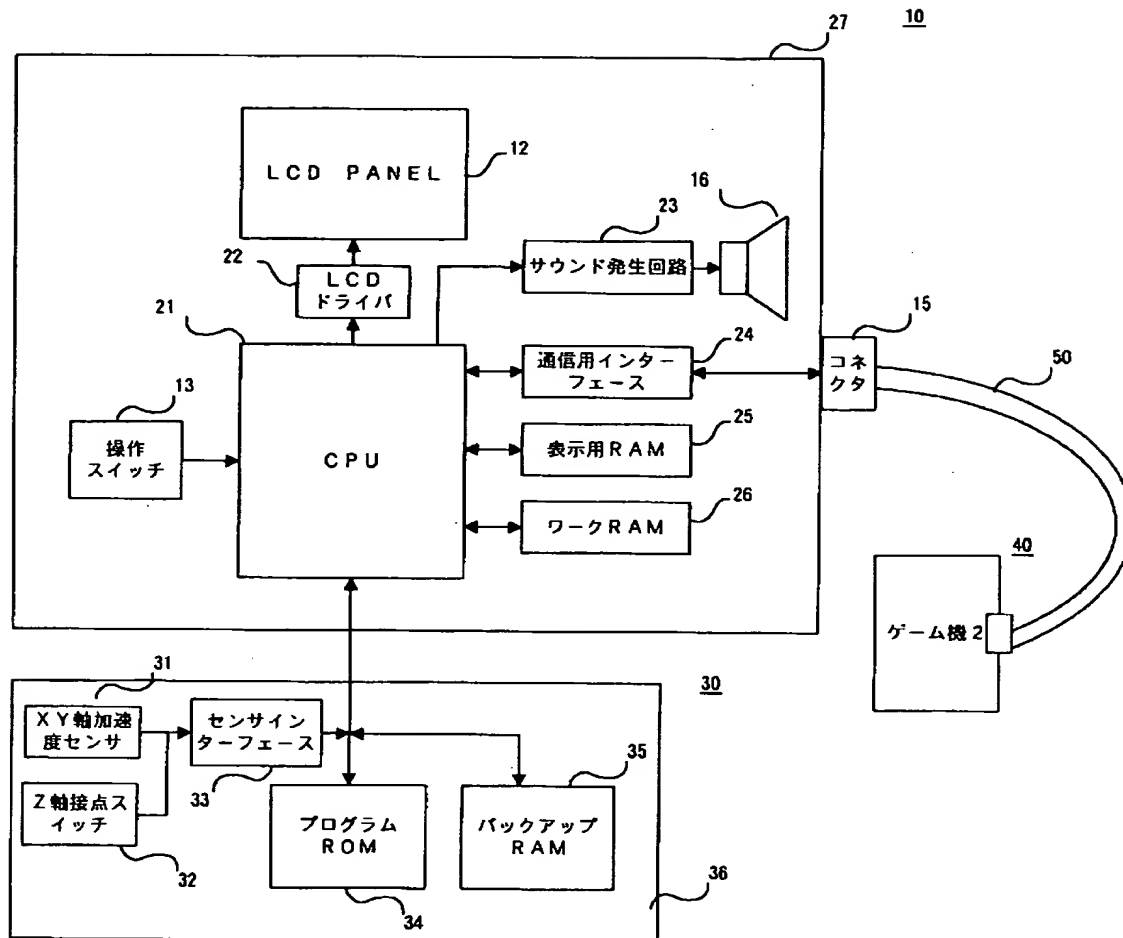
【図 1】



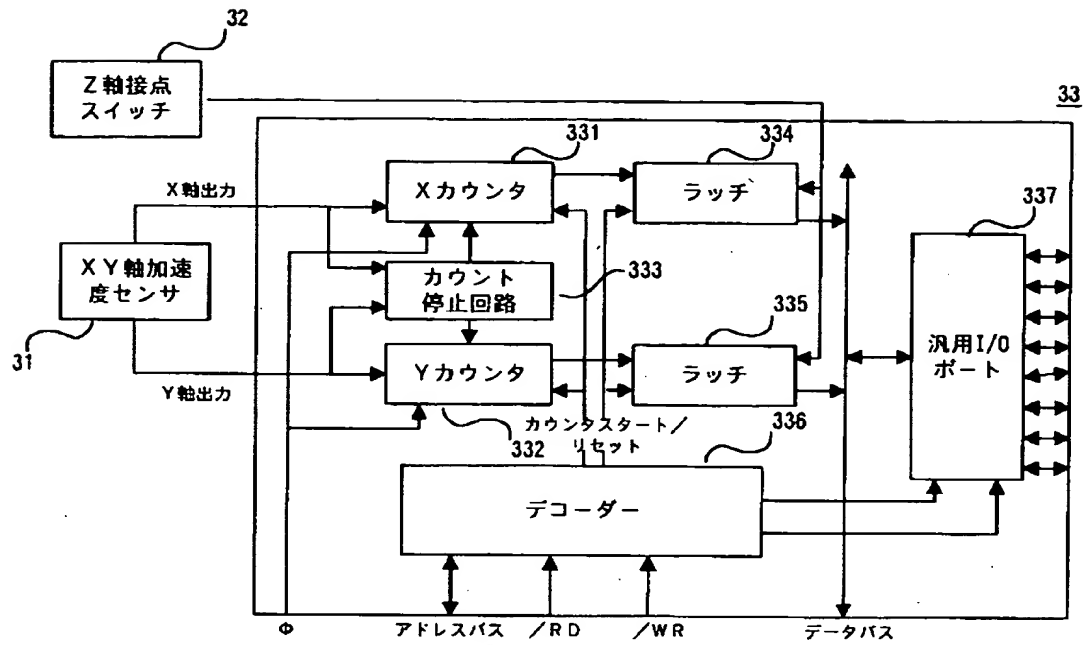
【図 2】



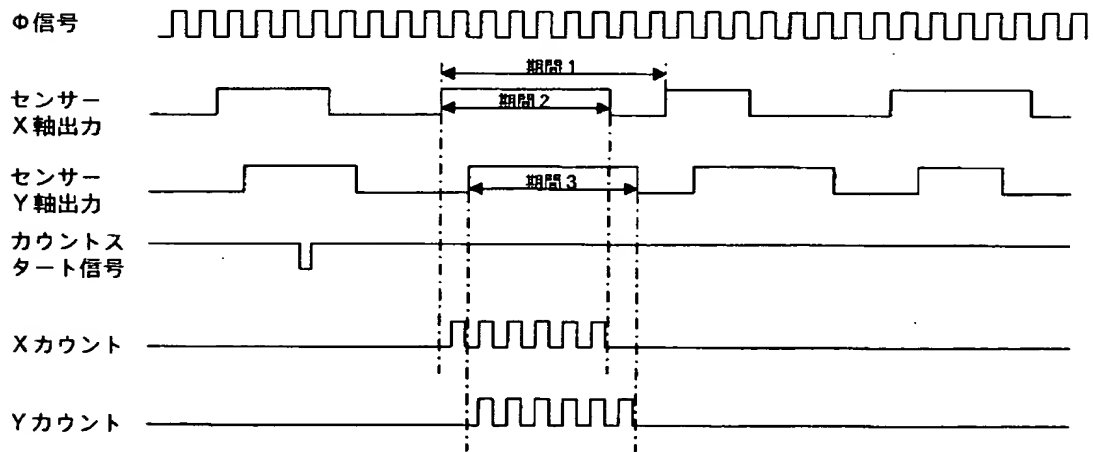
【図 3】



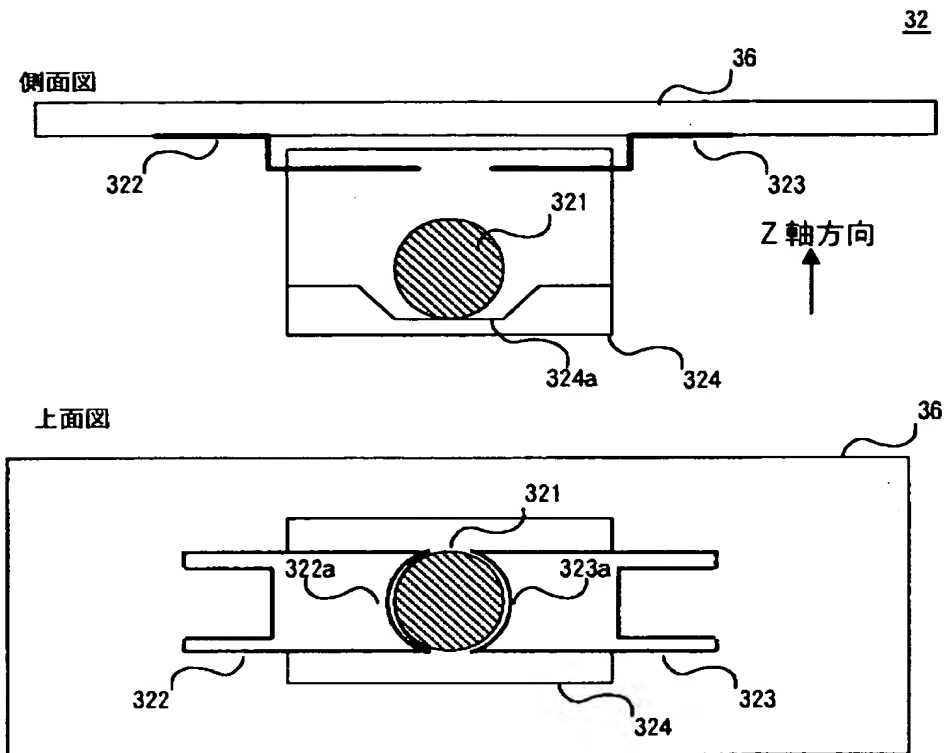
【図 4】



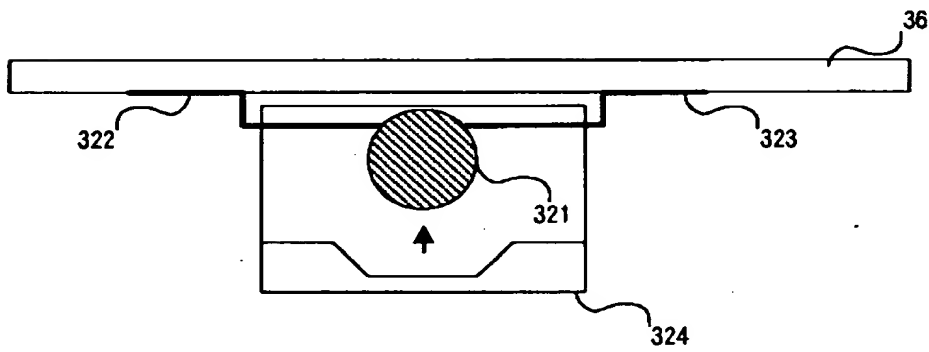
【図 5】



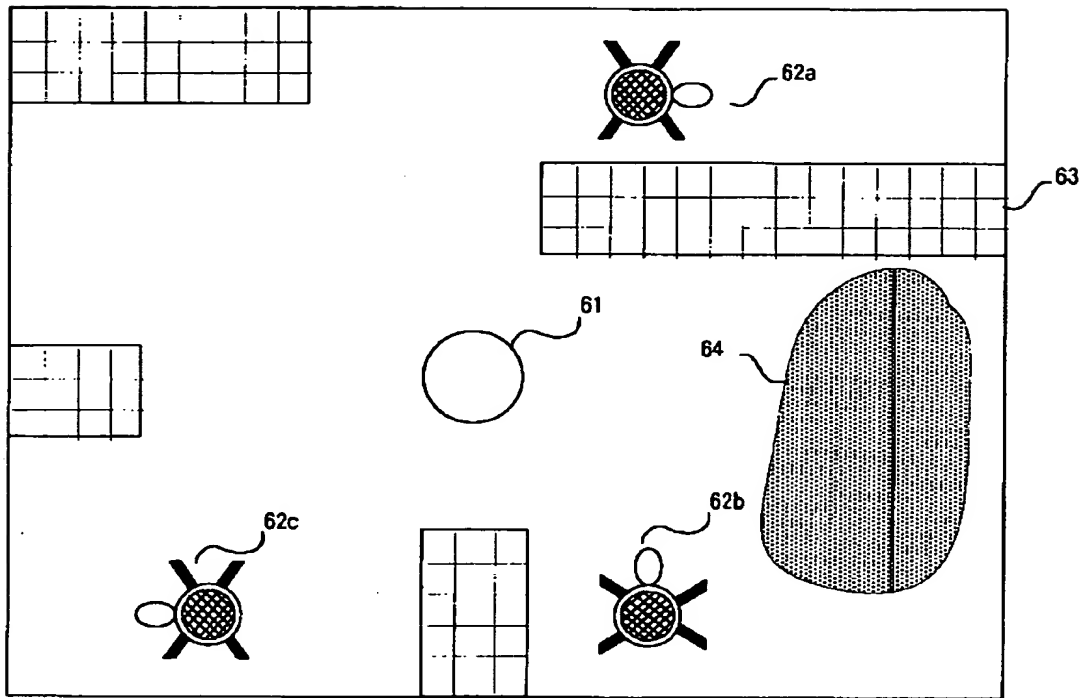
【図 6】



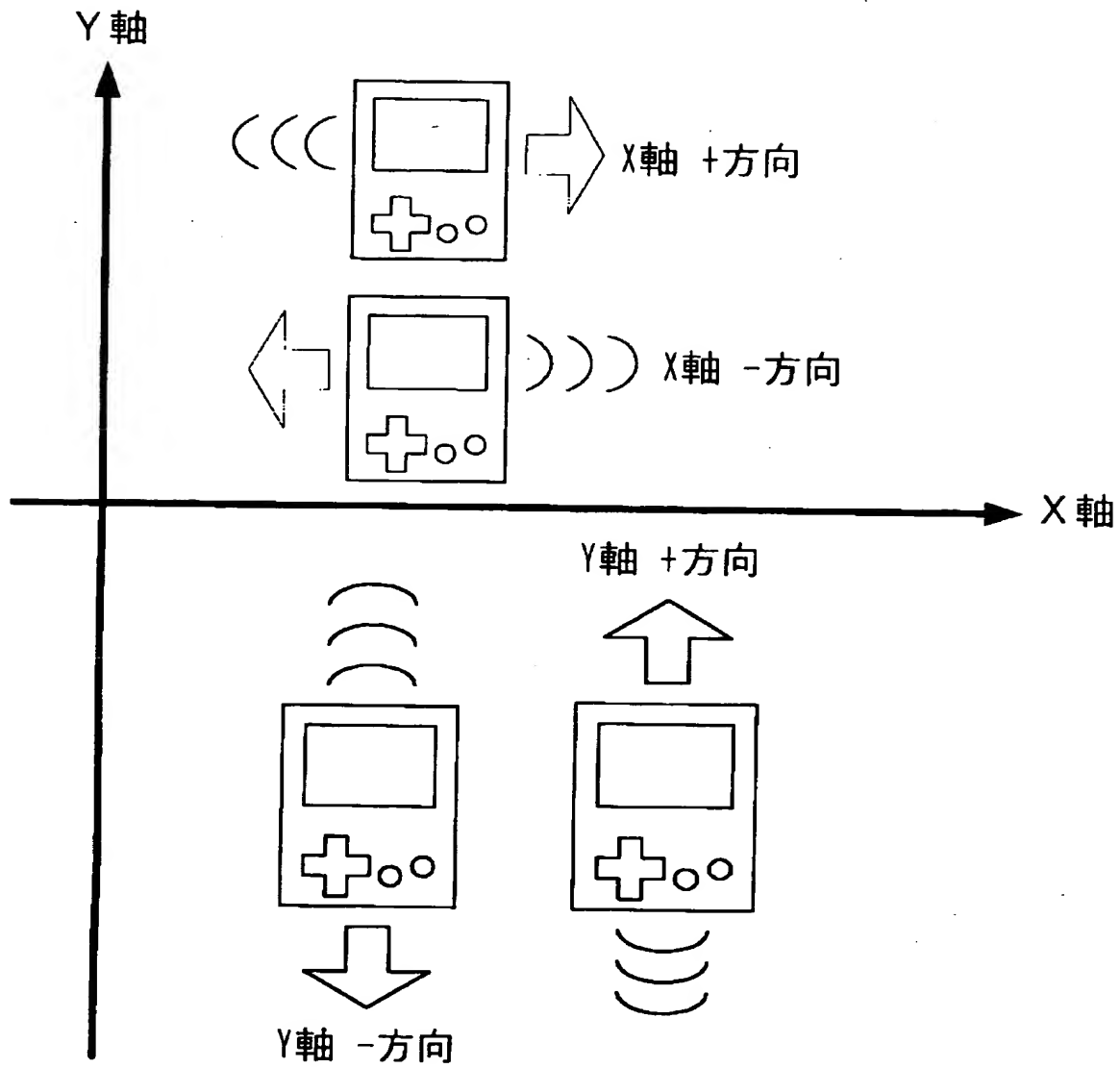
【図 7】



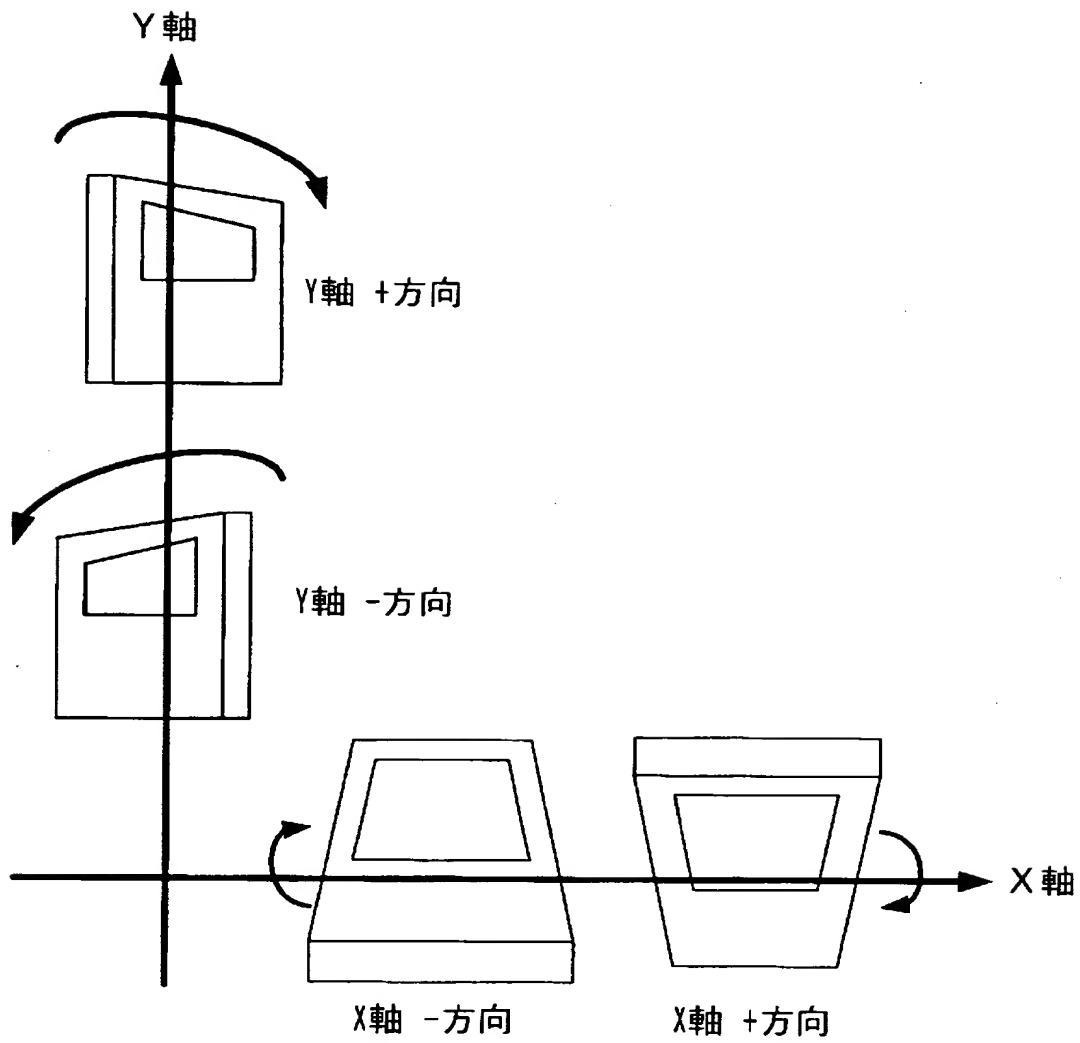
【図 8】



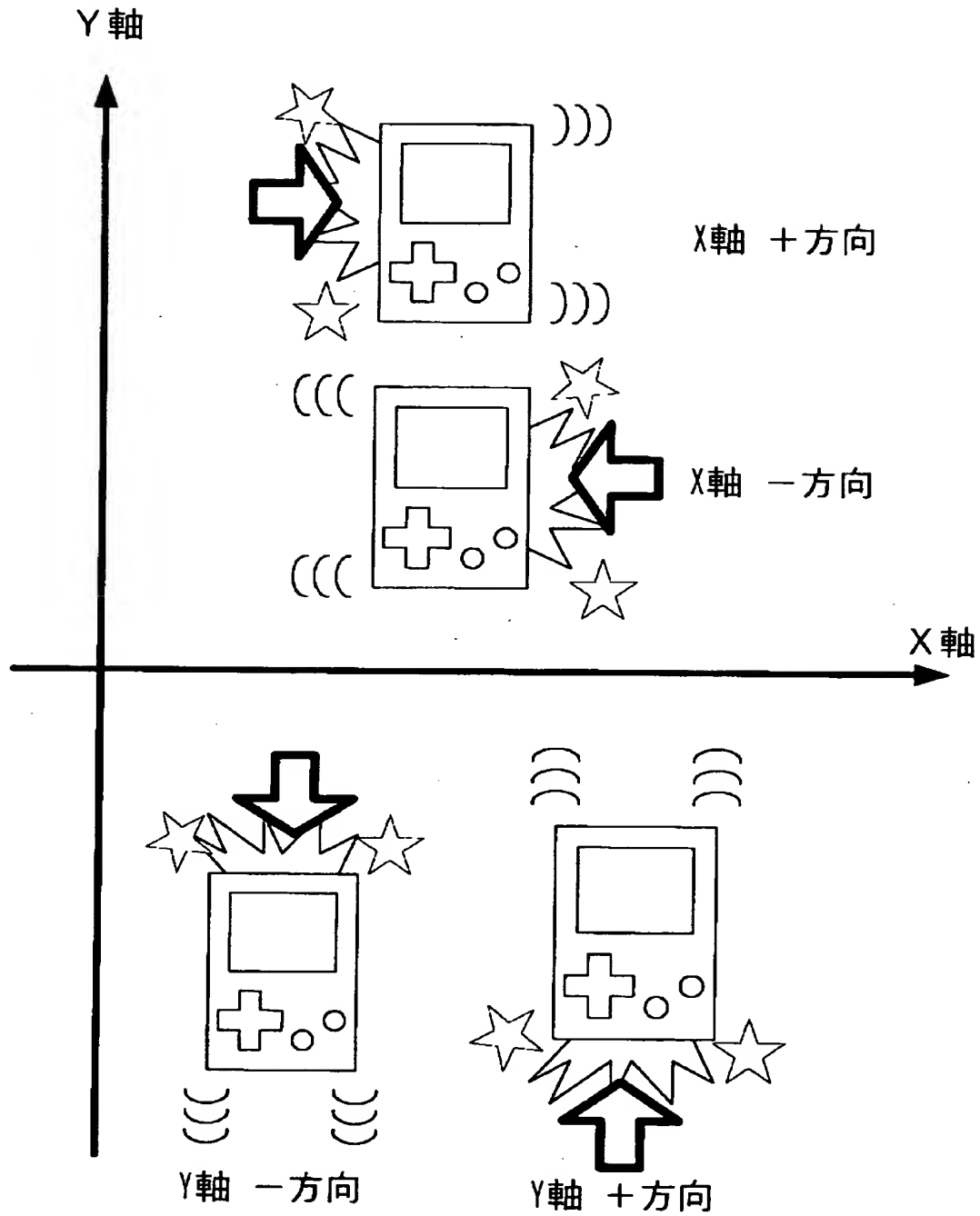
【図 9】



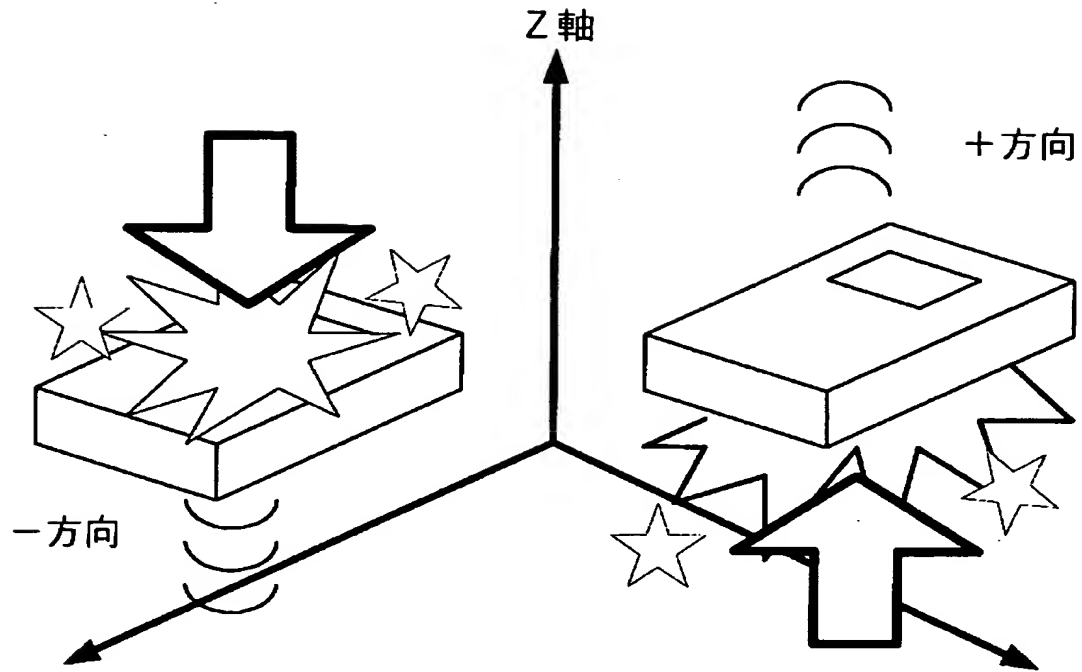
【図 1 0】



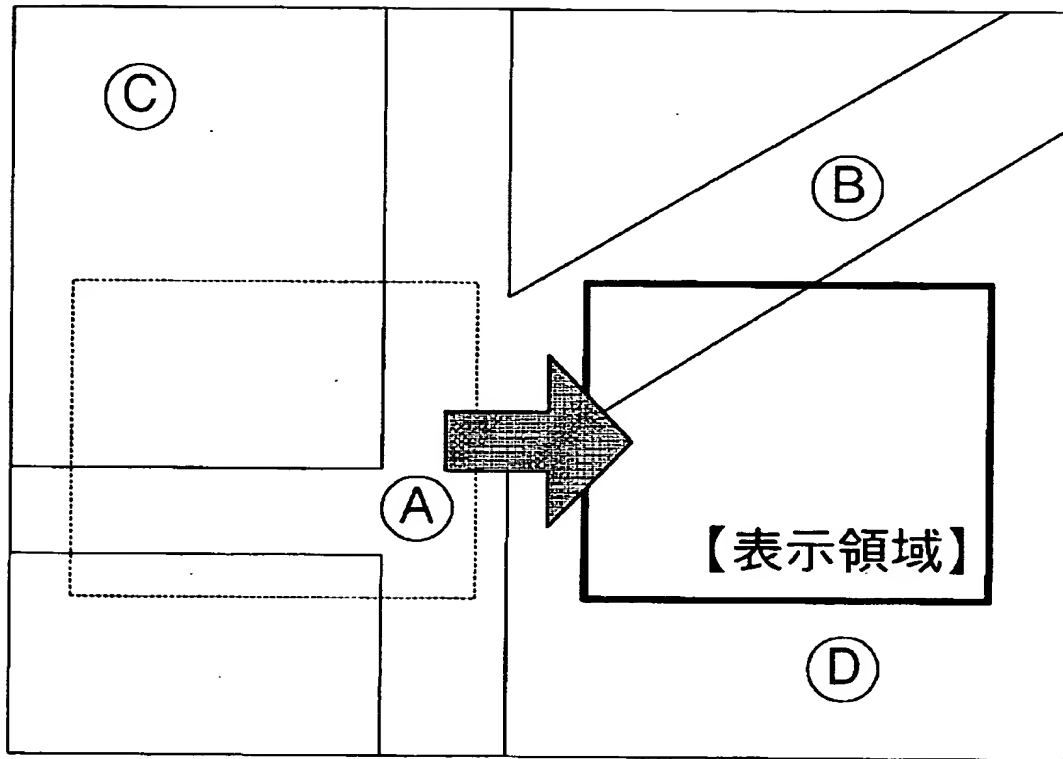
【図 11】



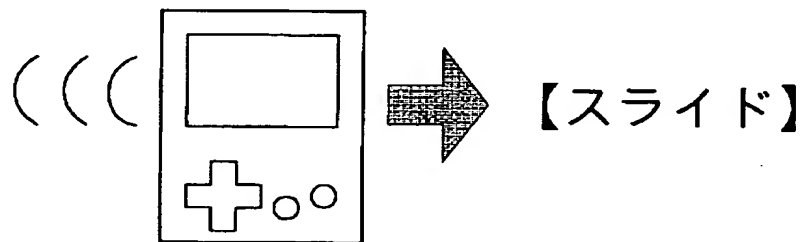
【図 1 2】



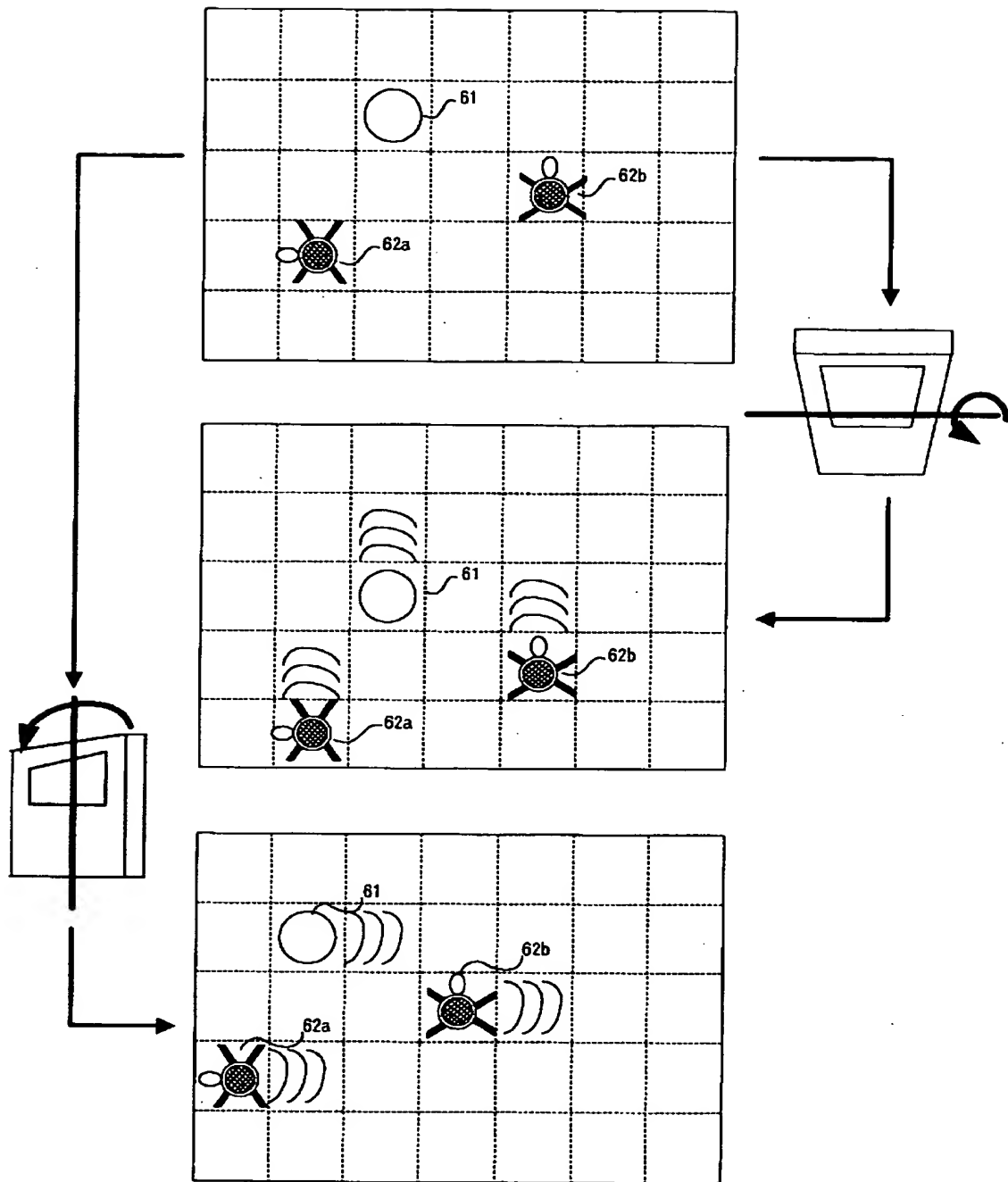
【図 13】



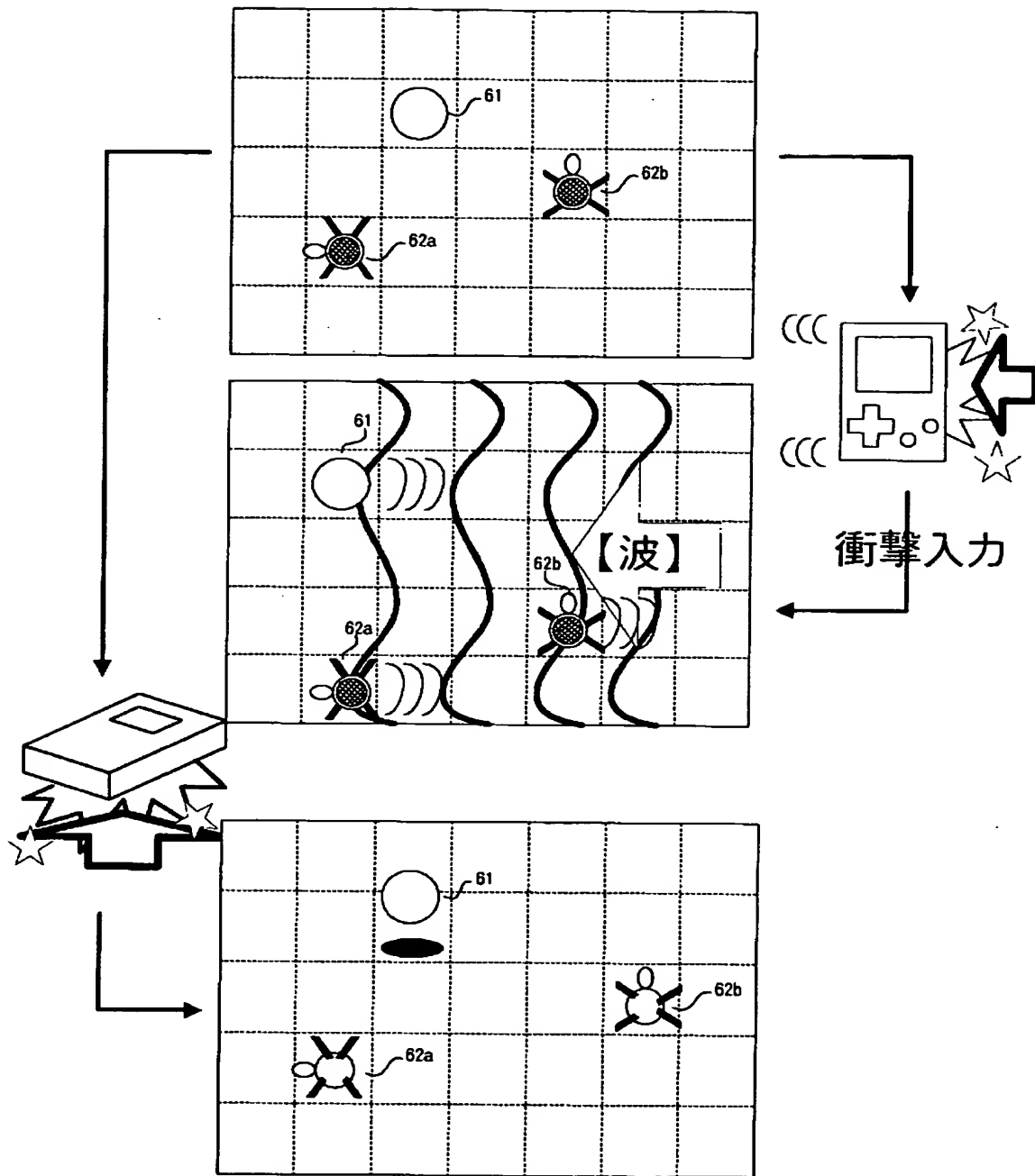
【仮想マップ】



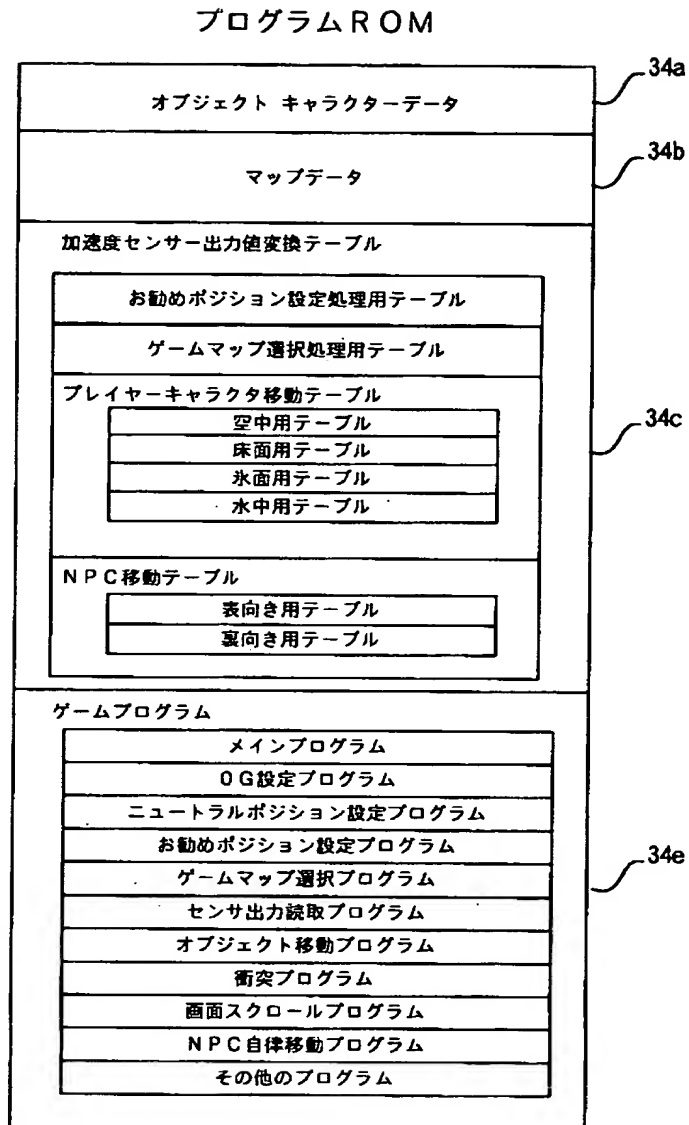
【図 1 4】



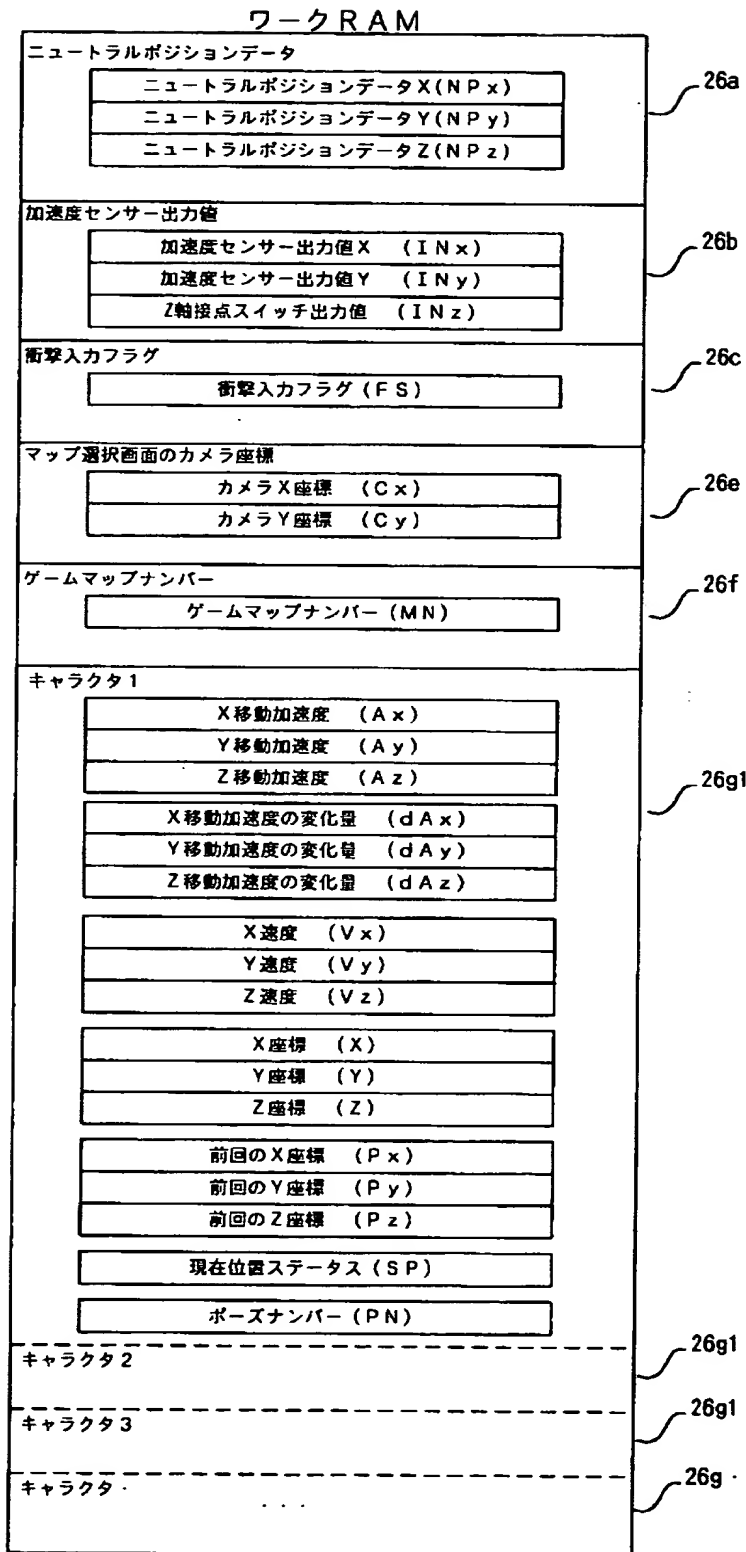
【図 15】



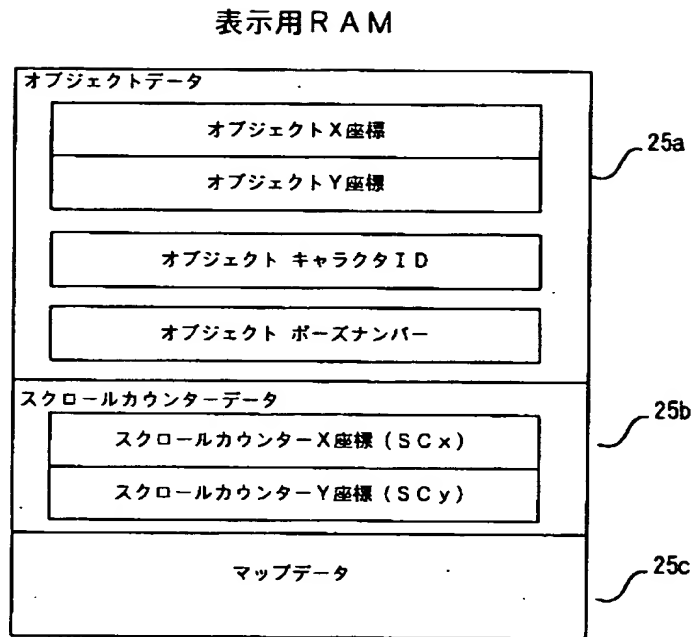
【図 1 6】



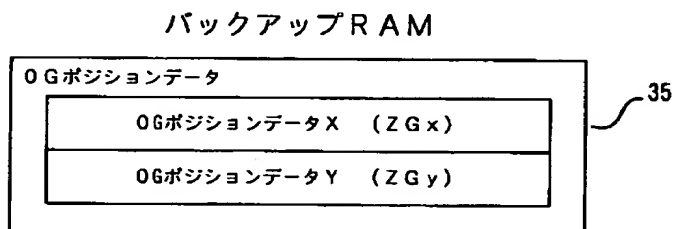
【図 1 7】



【図 1 8】



【図 1 9】



【図 2 0】

ゲームマップ選択処理用テーブル

	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	カメラ X 座 標 (Cx)の変化量	× 2	—	—	—	—
センサ出力値 X (INx)	カメラ Y 座 標 (Cy)の変化量	× 2	—	—	—	—
Z 軸接点スイッ チ出力値(INz)	マップ決定	—	—	—	—	—
衝撃入力フラグ (FS)	—	—	—	—	—	—

【図 2 1】

プレイヤーキャラクタ移動用テーブル（空中用）

	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	—	—	—	—	—	—
センサ出力値 X (INx)	—	—	—	—	—	—
Z 軸接点スイッ チ出力値(INz)	Z 移動加速度の 変化量(dAz)	× 1	—	—	—	—
衝撃入力フラグ (FS)	—	—	—	—	—	—

【図 2 2】

プレイヤーキャラクタ移動用テーブル（床面用）

	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	X 移動加速度の 変化量(dAx)	× 2	INx>20	40	—	—
センサ出力値 X (INx)	Y 移動加速度の 変化量(dAy)	× 2	INx>20	40	—	—
Z 軸接点スイッ チ出力値(INz)	Z 移動加速度の 変化量(dAz)	× 1	—	—	—	—
衝撃入力フラグ (FS)	X,Y 移動加速度 の変化量 (dAx,dAy)	× 3	—	—	—	—

【図 2 3】

プレイヤーキャラクタ移動用テーブル（氷面用）

	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	X 移動加速度の 変化量(dAx)	× 3	INx>20	60	—	—
センサ出力値 X (INx)	Y 移動加速度の 変化量(dAy)	× 3	INy>20	60	—	—
Z 軸接点スイッ チ出力値(INz)	Z 移動加速度の 変化量(dAz)	× 1	—	—	—	—
衝撃入力フラグ (FS)	X, Y 移動加速度 の変化量 (dAx, dAy)	× 5	—	—	—	—

【図 2 4】

プレイヤーキャラクタ移動用テーブル（水中用）

	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	X 移動加速度の 変化量(dAx)	× 1/2	INx>10	5	—	—
センサ出力値 X (INx)	Y 移動加速度の 変化量(dAy)	× 1/2	INy>10	5	—	—
Z 軸接点スイッ チ出力値(INz)	Z 移動加速度の 変化量(dAz)	× 1	—	—	—	—
衝撃入力フラグ (FS)	—	—	—	—	—	—

【図 2 5】

NPC 移動用テーブル（亀表向き用）

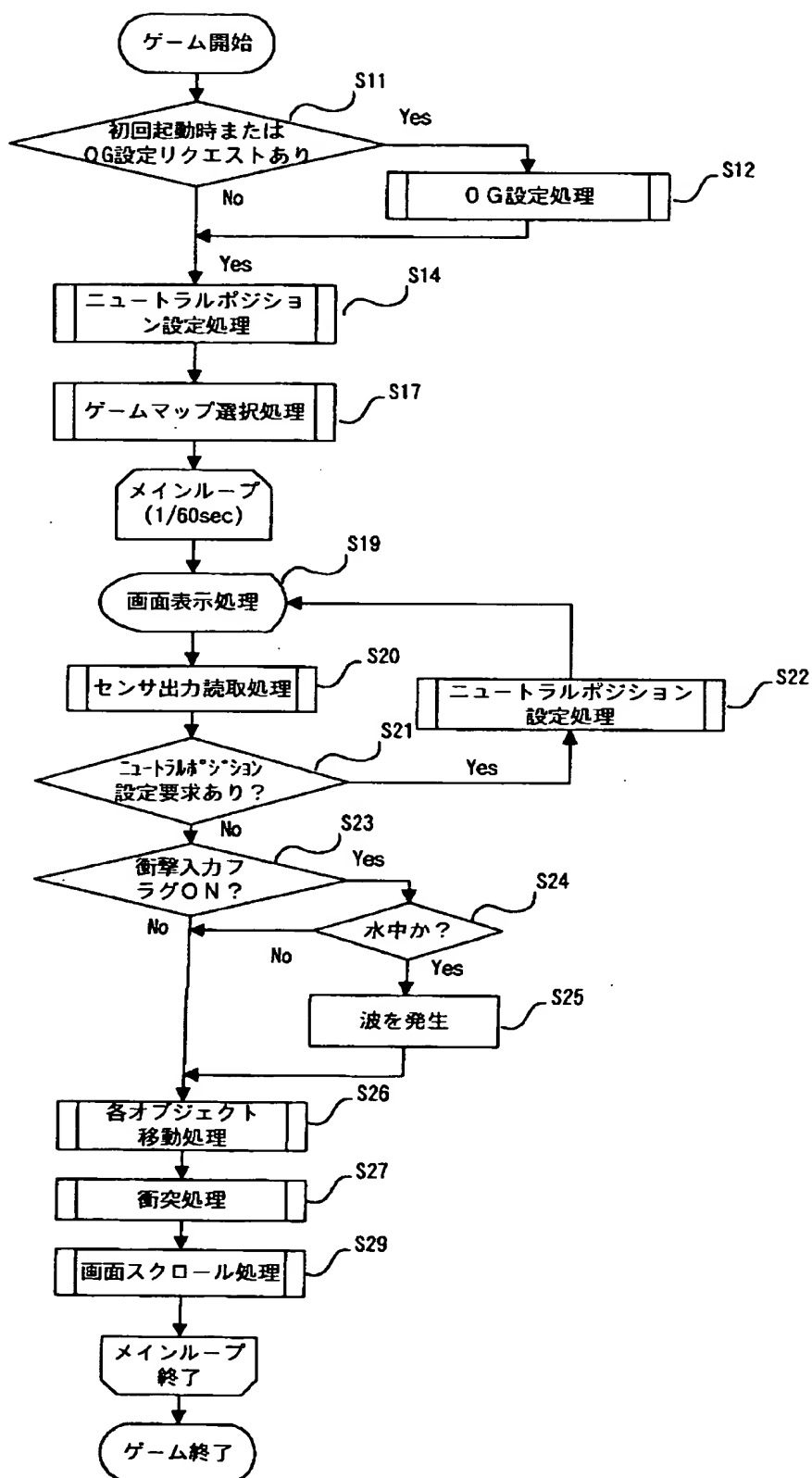
	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	X 移動加速度の 変化量(dAx)	× 1/2	INx<10	0	INx>20	10
センサ出力値 X (INx)	Y 移動加速度の 変化量(dAy)	× 1/2	INy<10	0	INx>20	10
Z 軸接点スイッ チ出力値(INz)	表裏逆転	—	—	—	—	—
衝撃入力フラグ (FS)	—	—	—	—	—	—

【図 2 6】

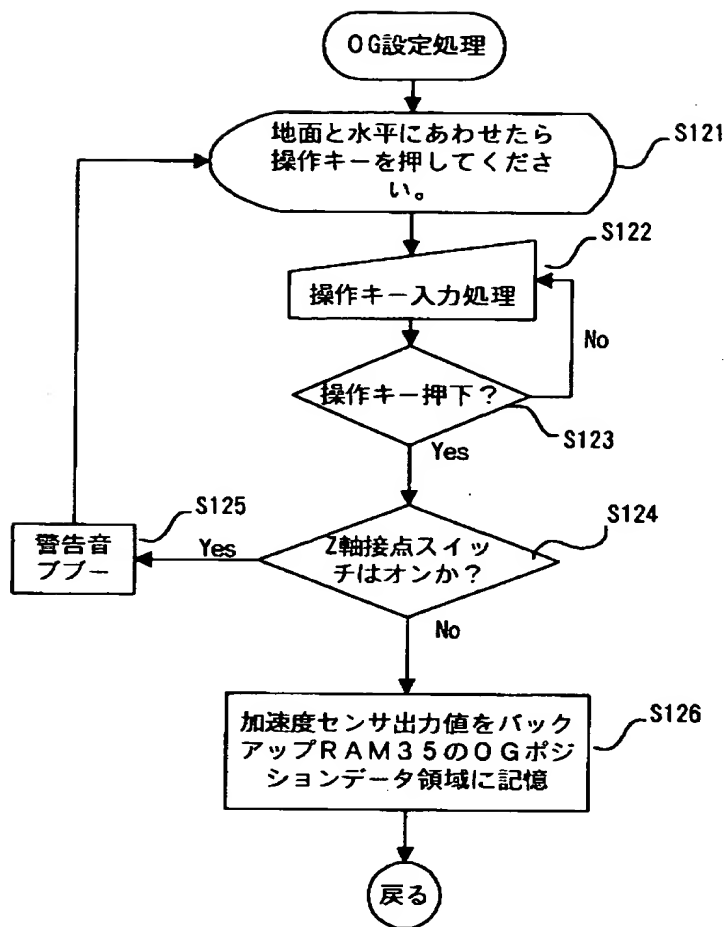
N P C 移動用テーブル（亀裏向き用）

	利用方法	補 正 比率	特殊補正 条件 1	特殊補正 数 1	特殊補正 条件 2	特殊補正 数 2
センサ出力値 X (INx)	X 移動加速度の 変化量(dAx)	× 2	INx>20	40	—	—
センサ出力値 X (INx)	Y 移動加速度の 変化量(dAy)	× 2	INy>20	40	—	—
Z 軸接点スイッ チ出力値(INz)	表裏逆転	—	—	—	—	—
衝撃入力フラグ (FS)	—	—	—	—	—	—

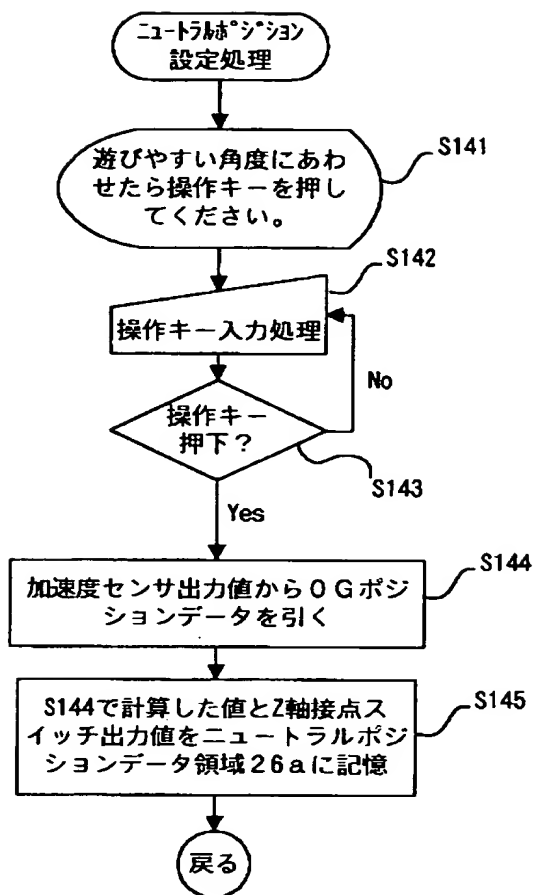
【図 27】



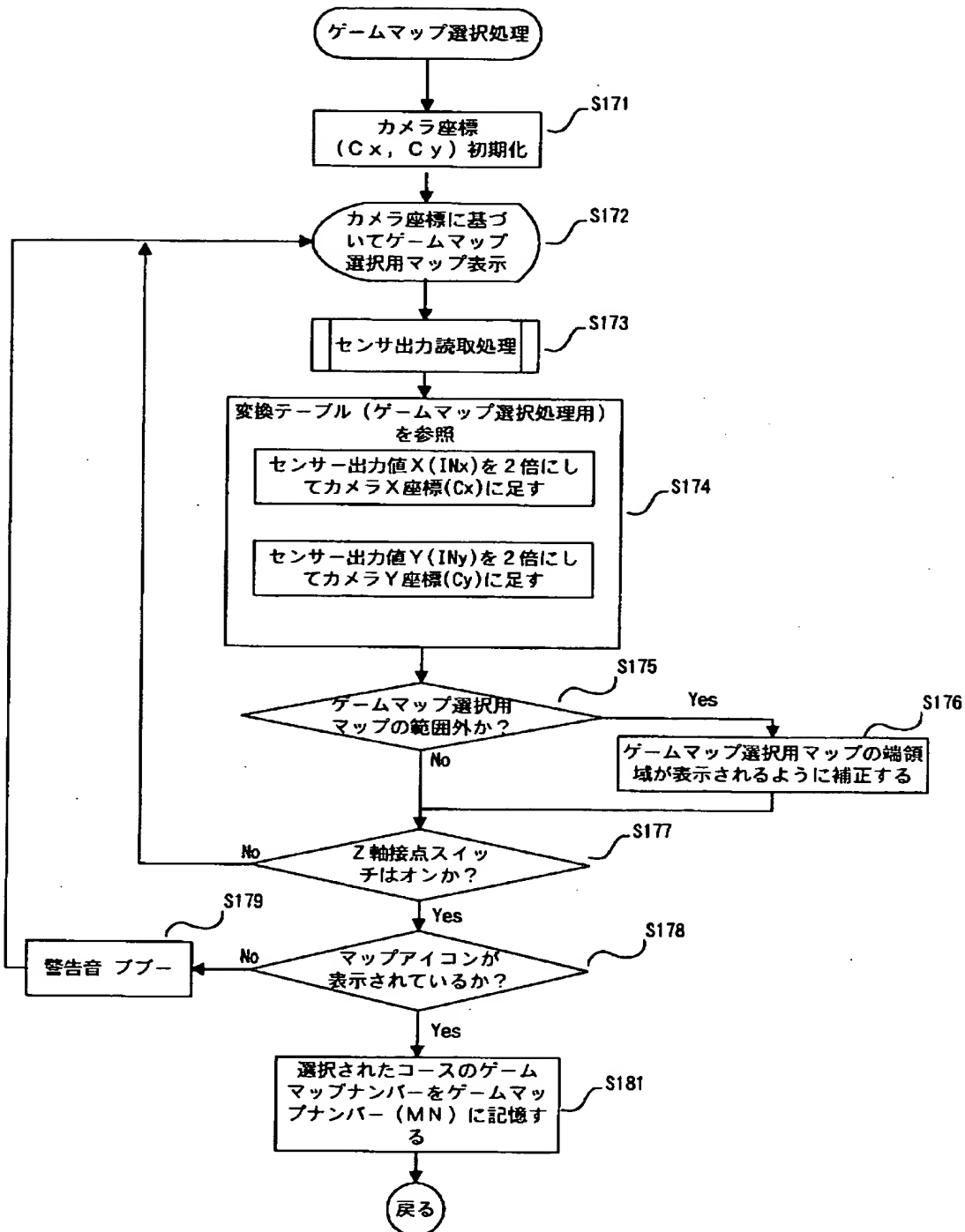
【図 2 8】



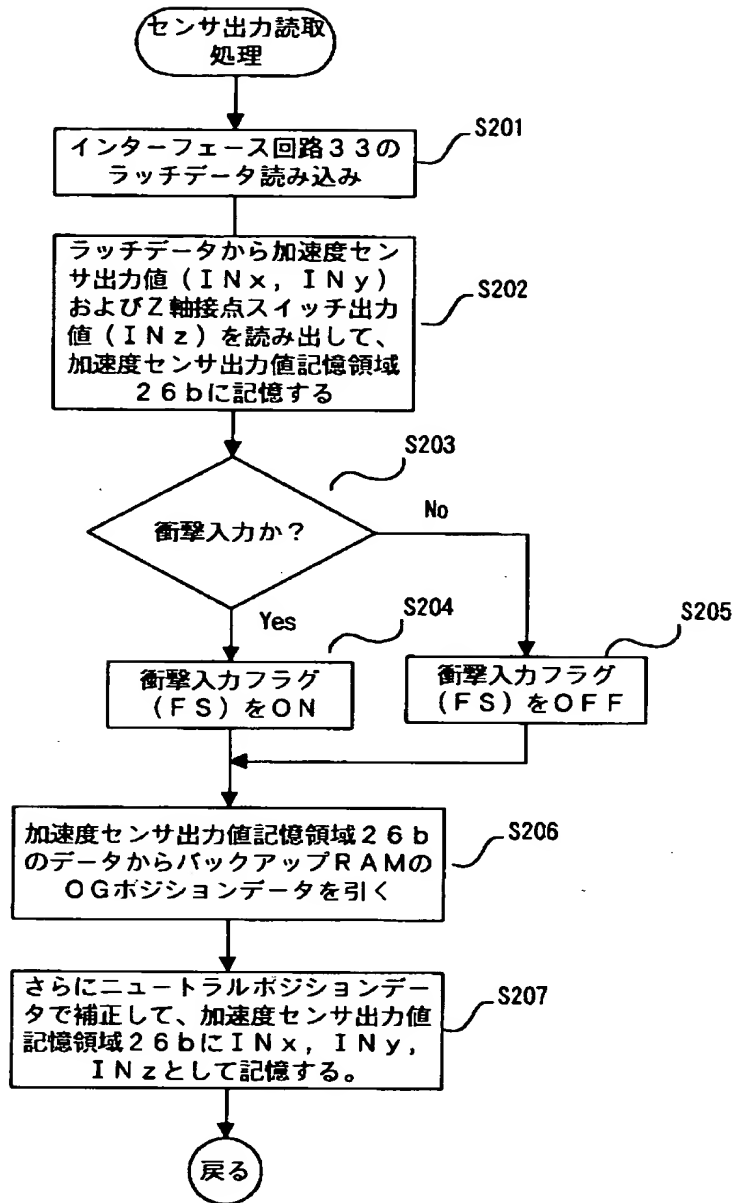
【図 2 9】



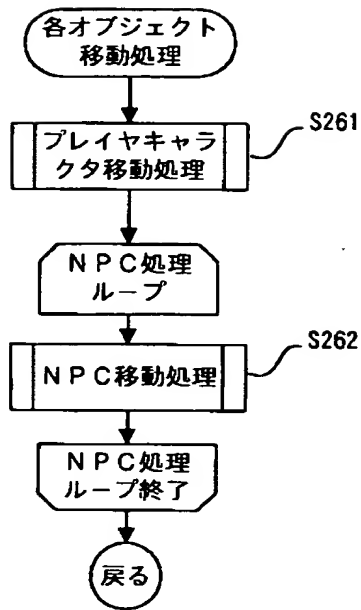
【図 30】



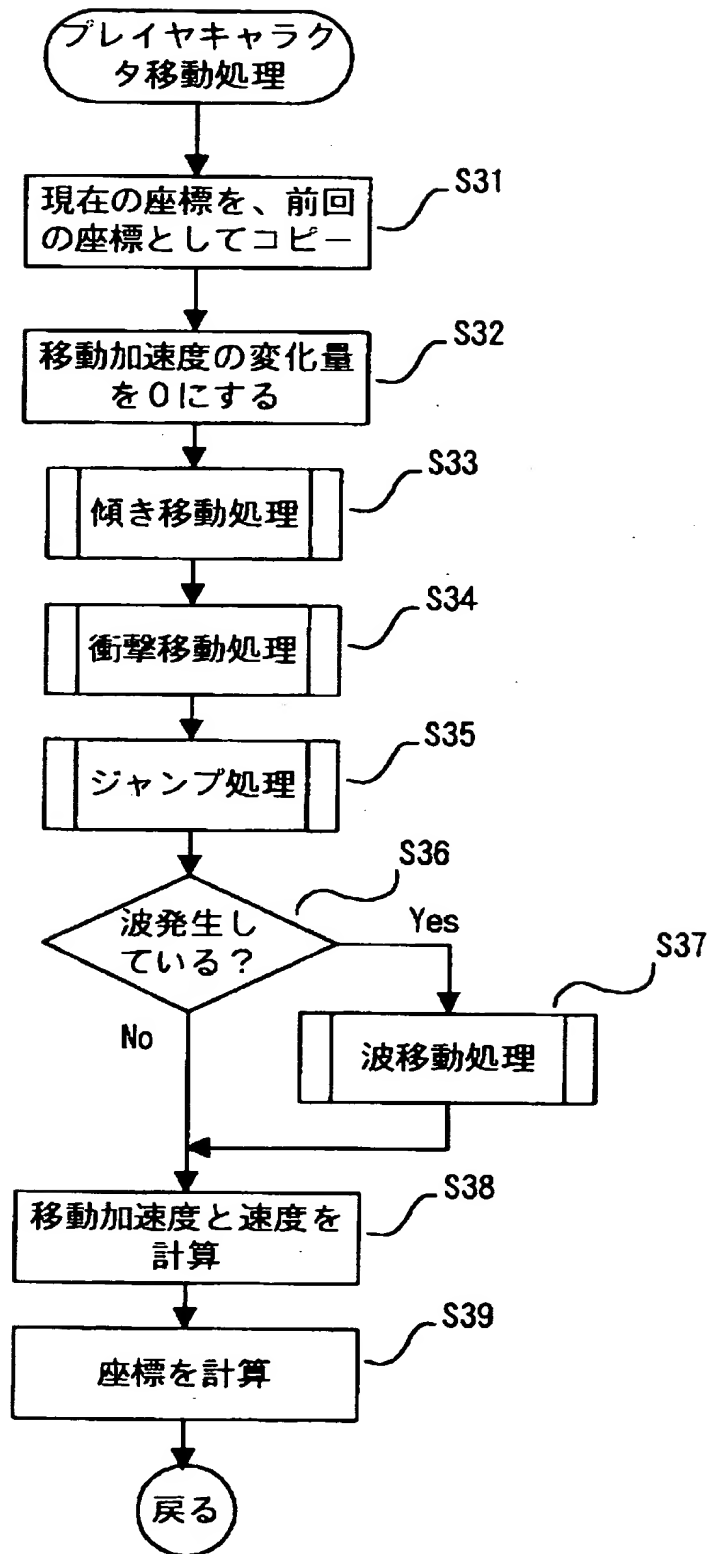
【図 31】



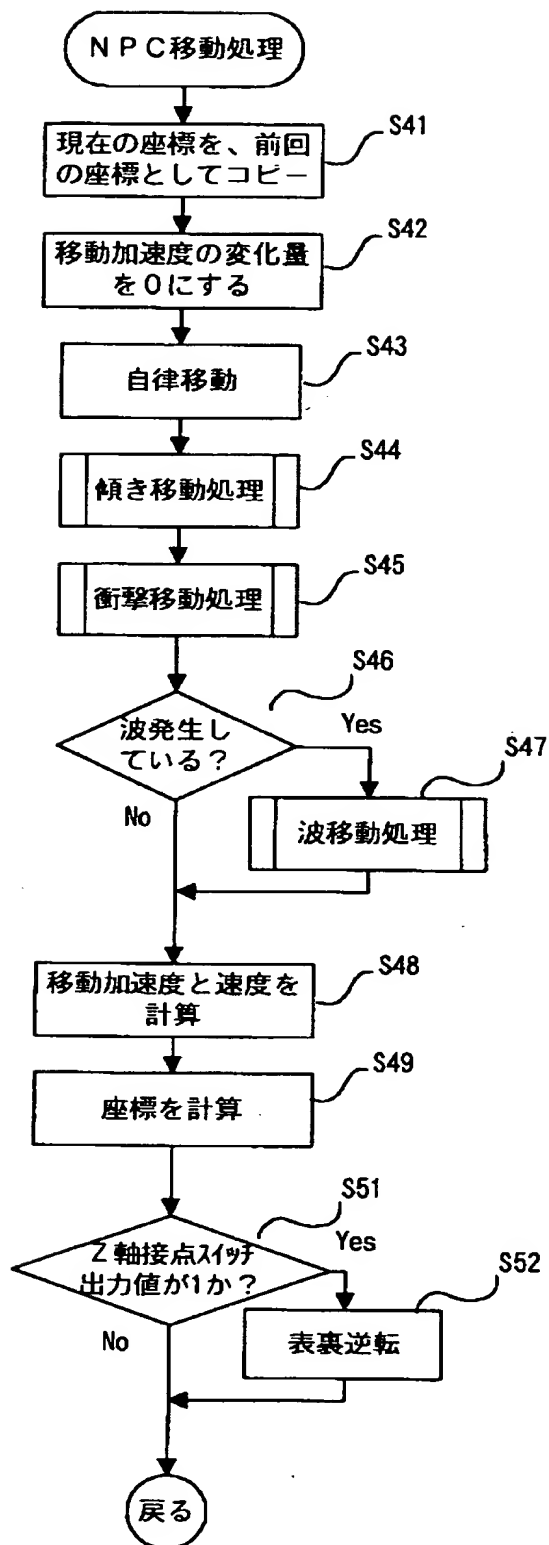
【図 3 2】



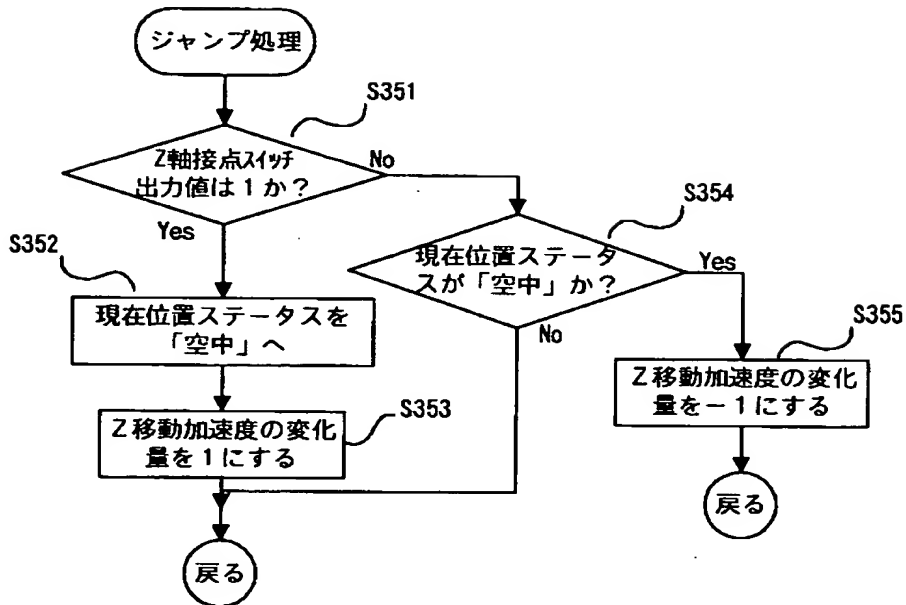
【図 3 3】



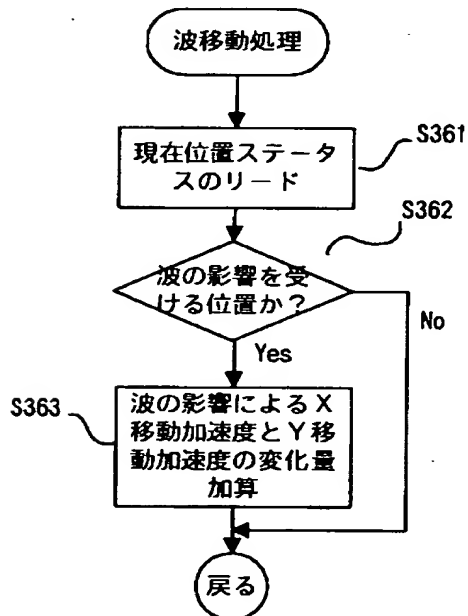
【図 3 4】



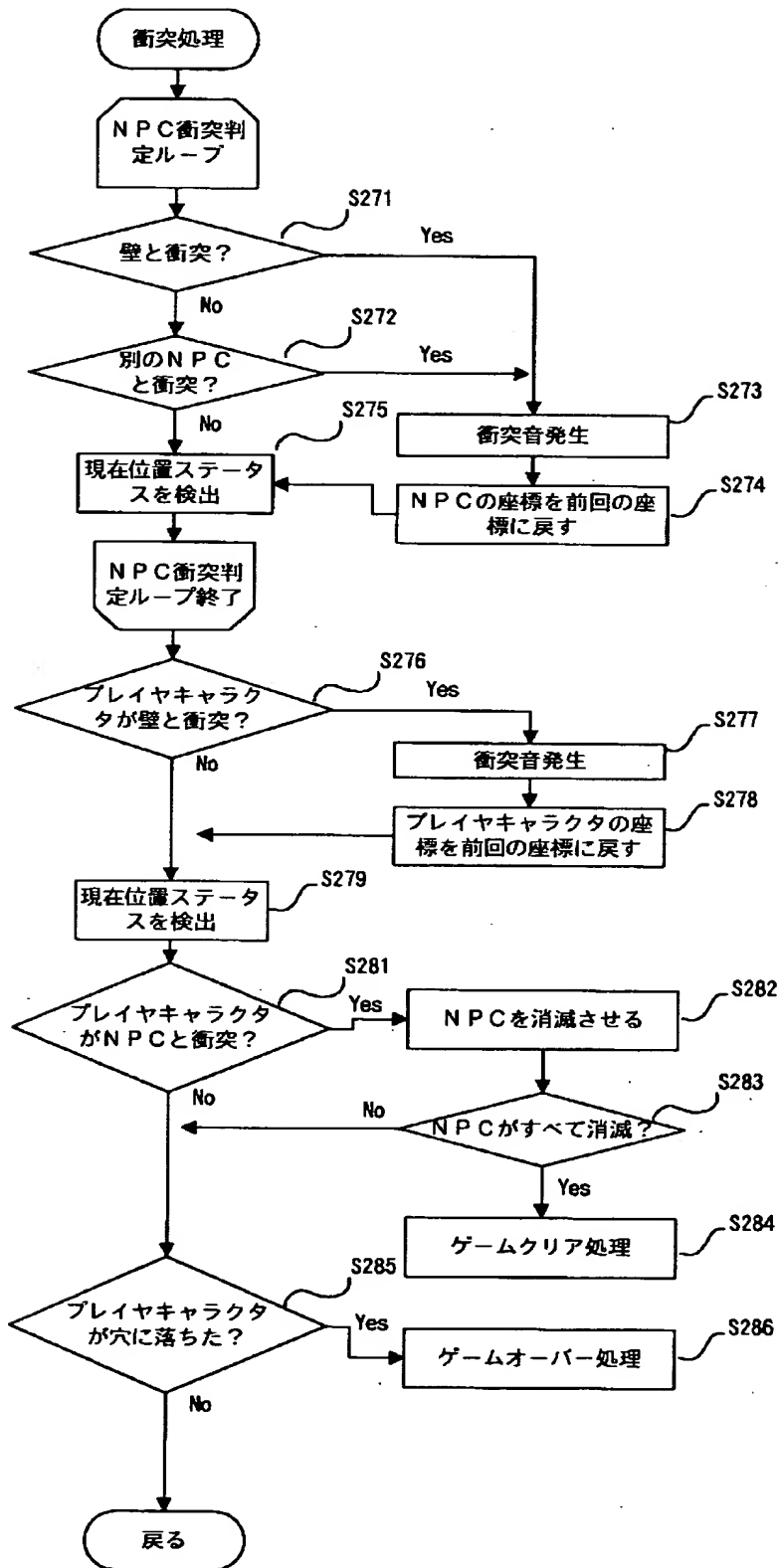
【図 3 5】



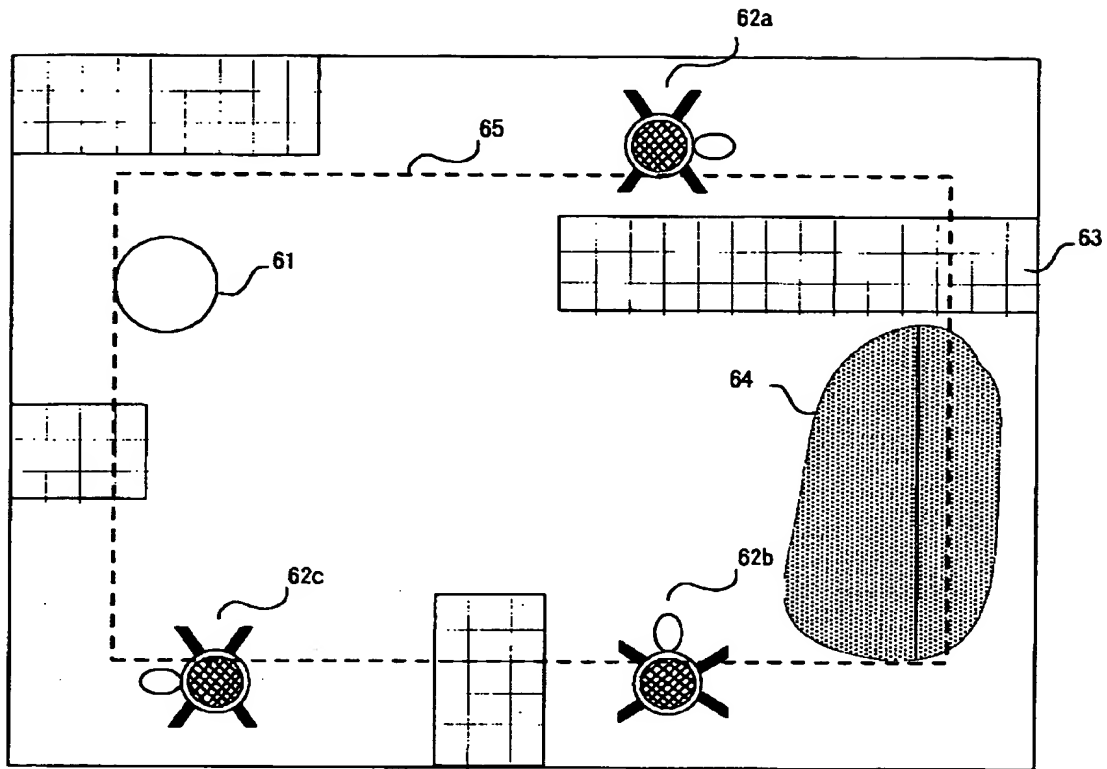
【図 3 6】



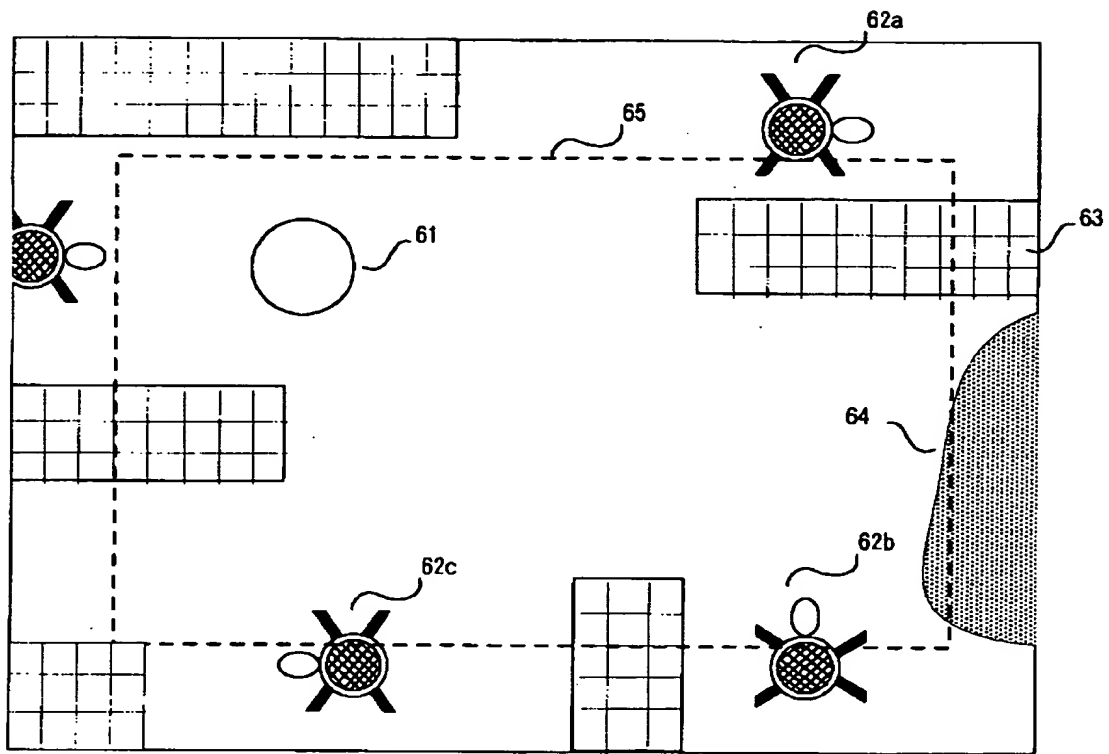
【図 37】



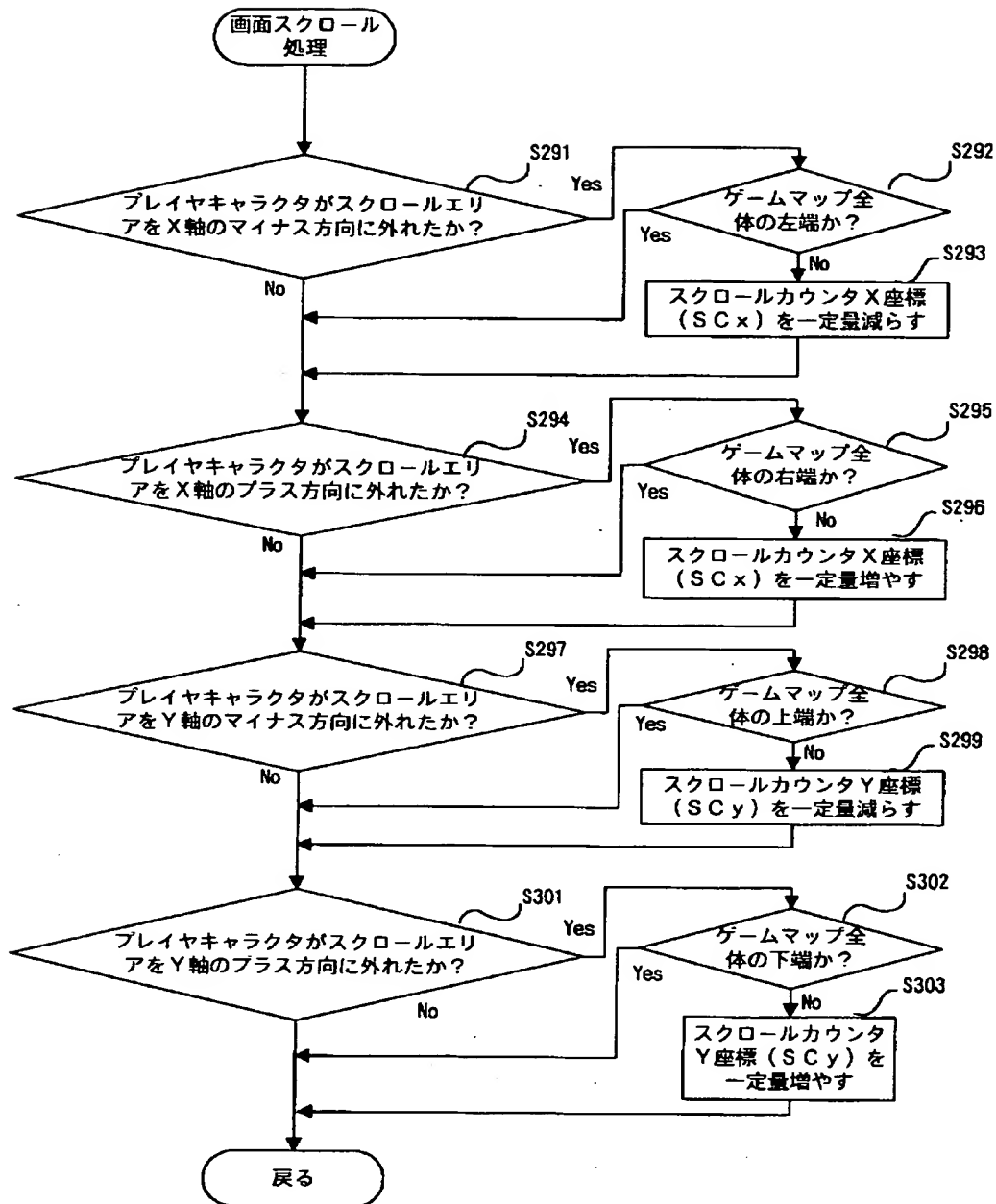
【図 3 8】



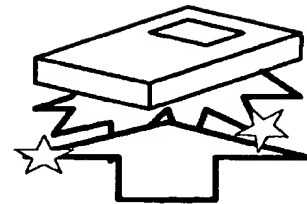
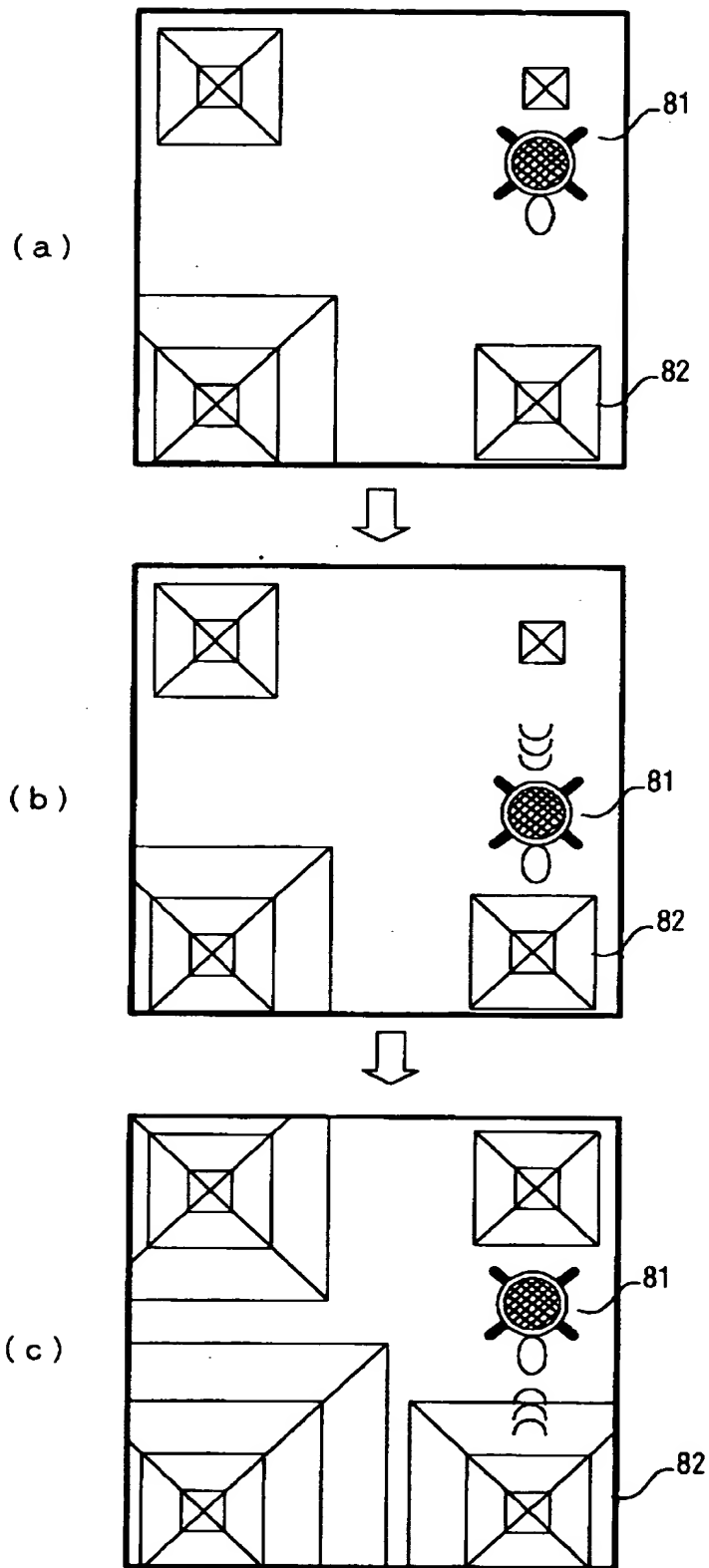
【図 3 9】



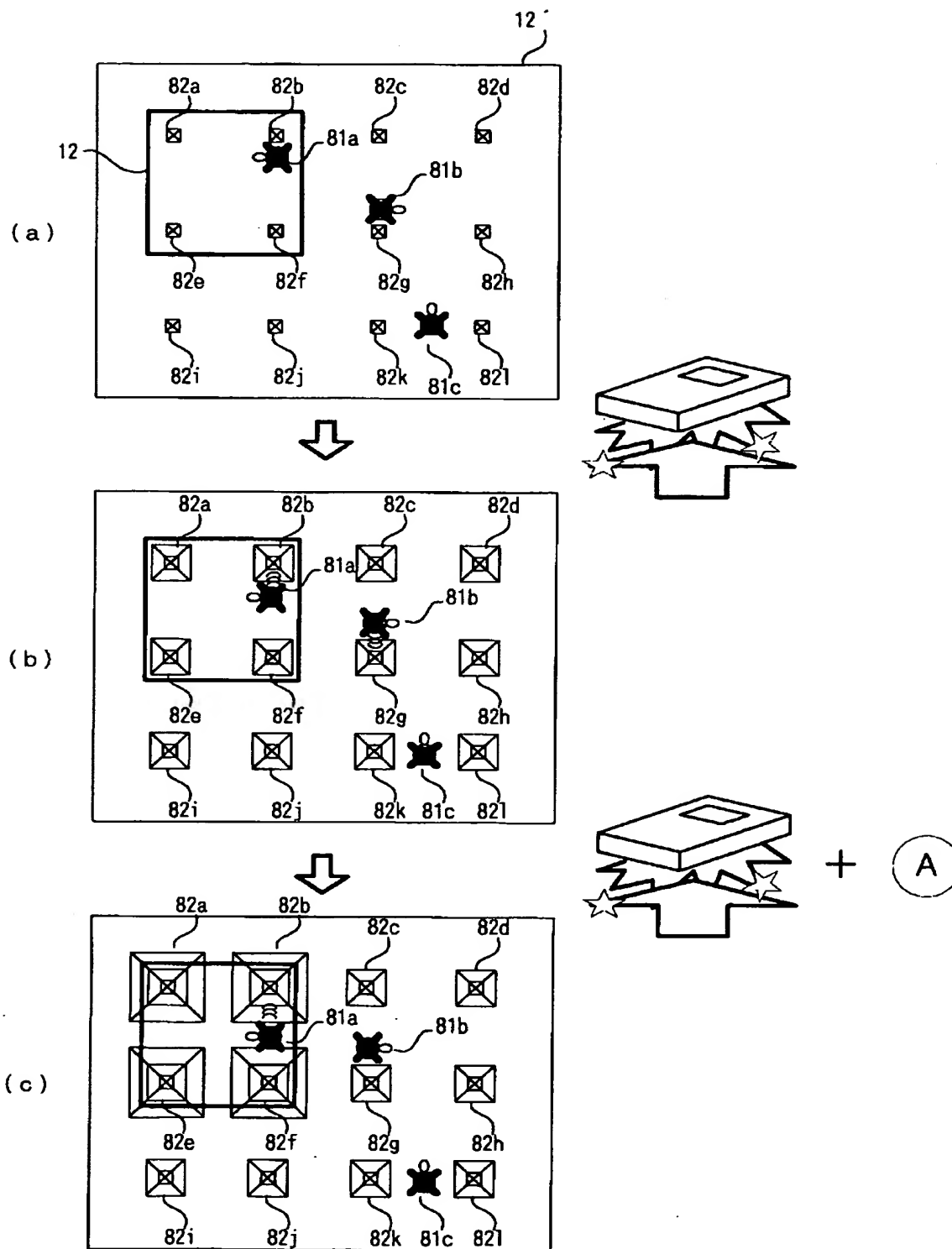
【図 40】



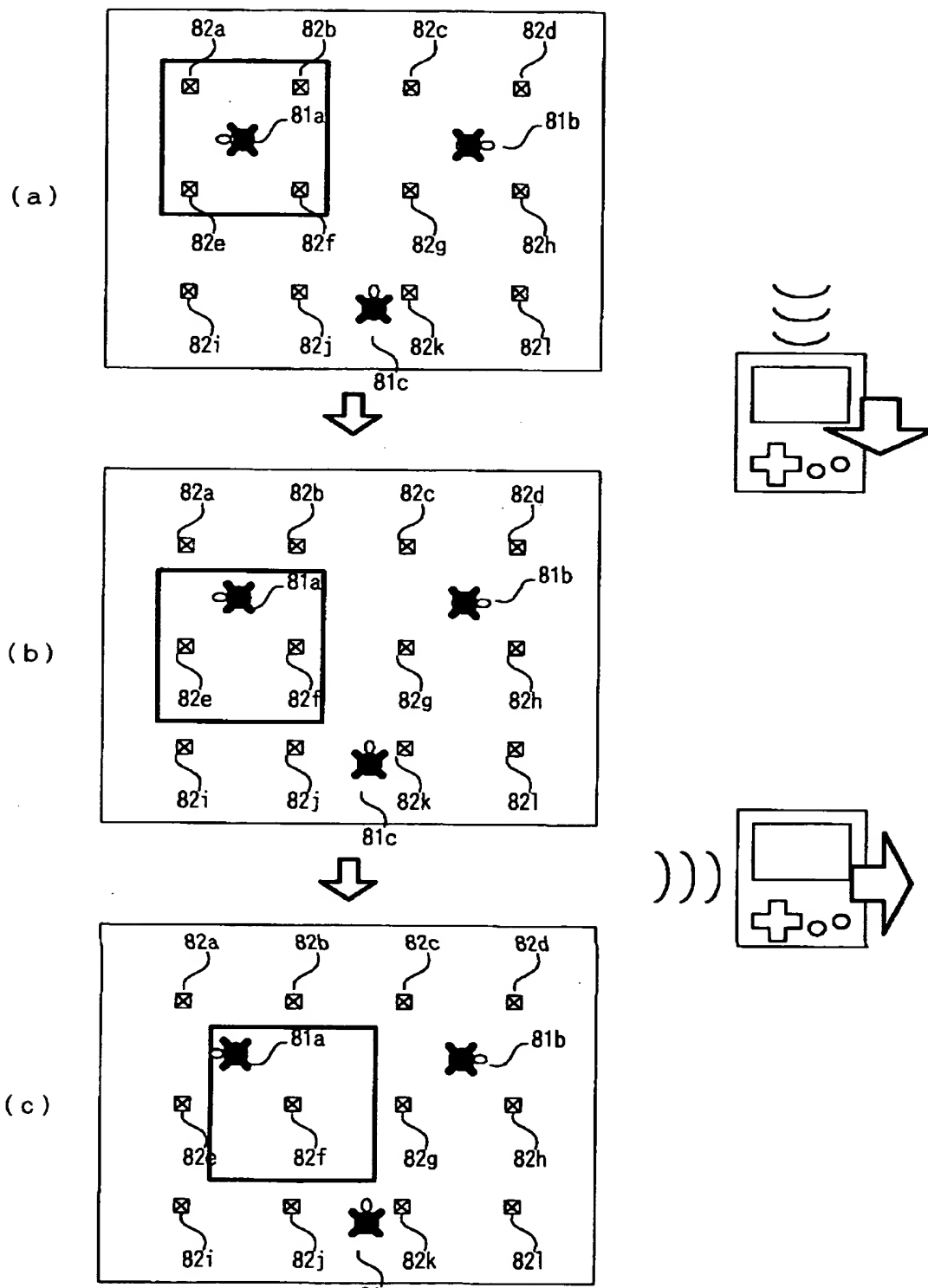
【図 4 1】



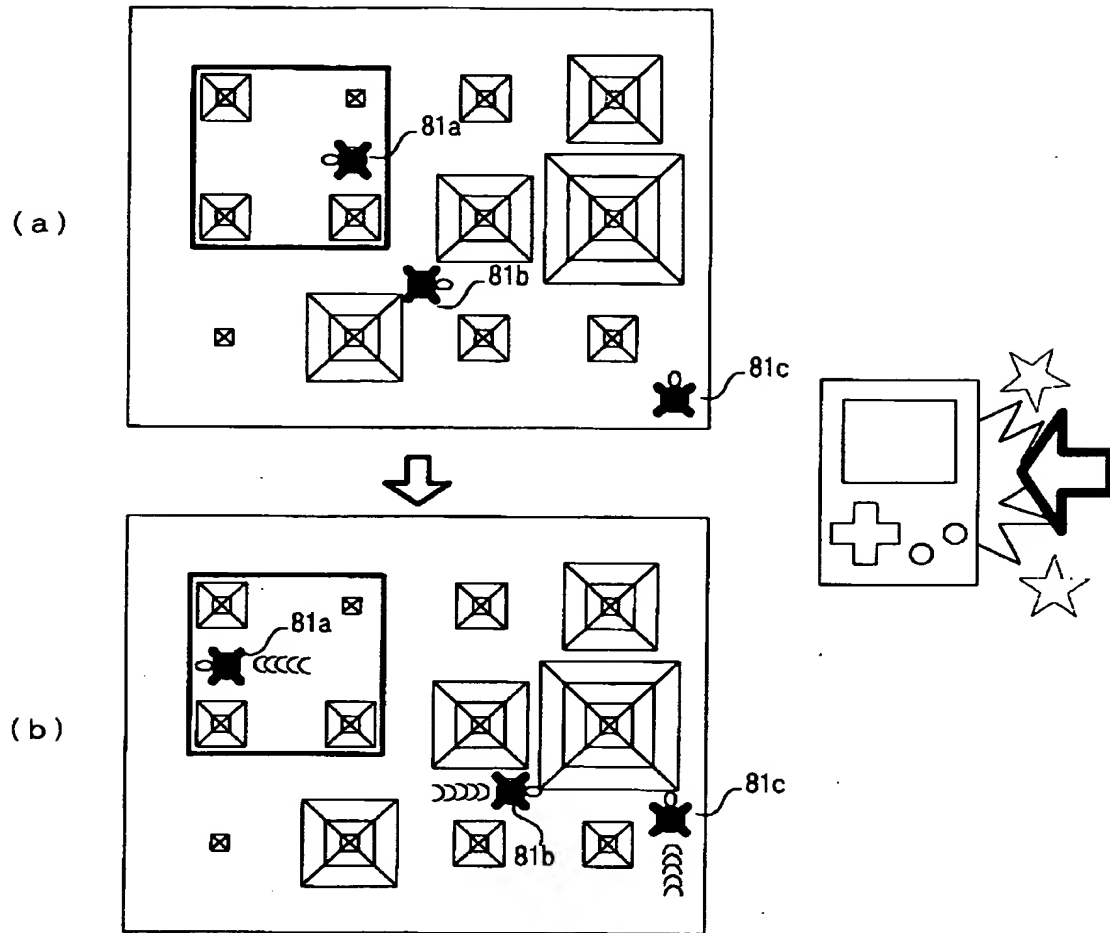
【図 4 2】



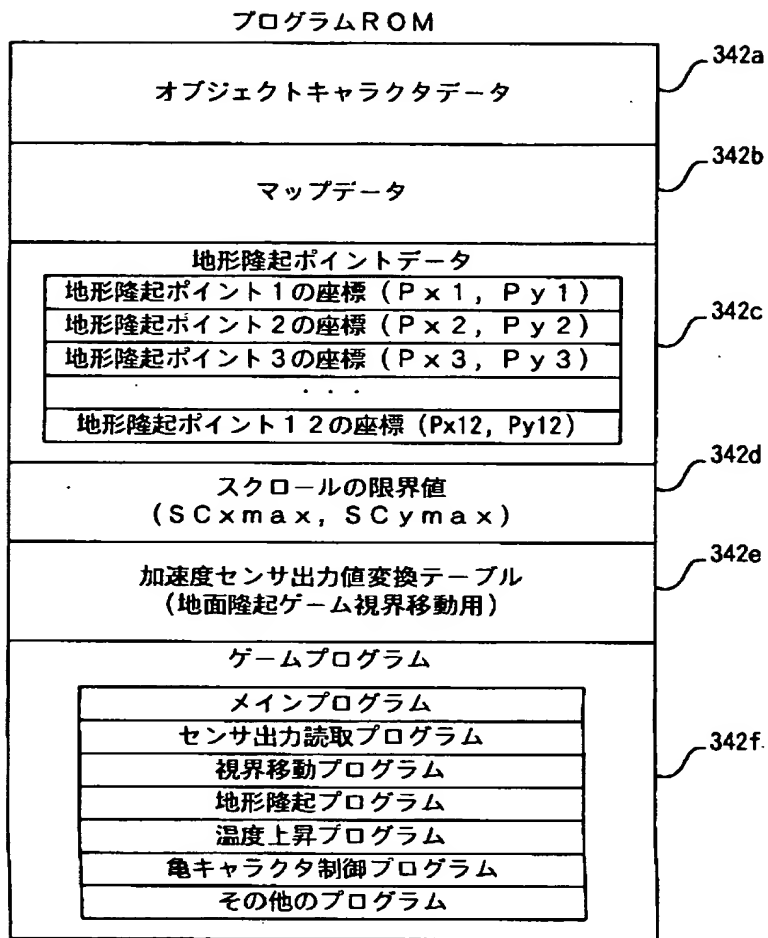
【図 4 3】



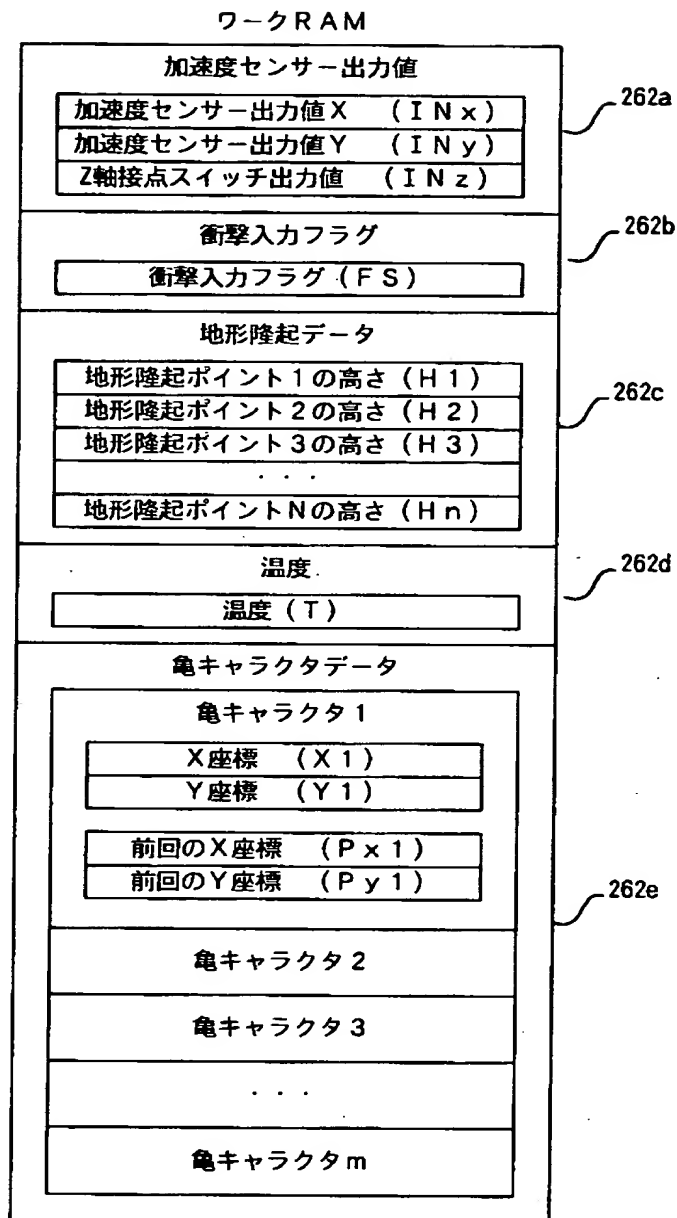
【図 4 4】



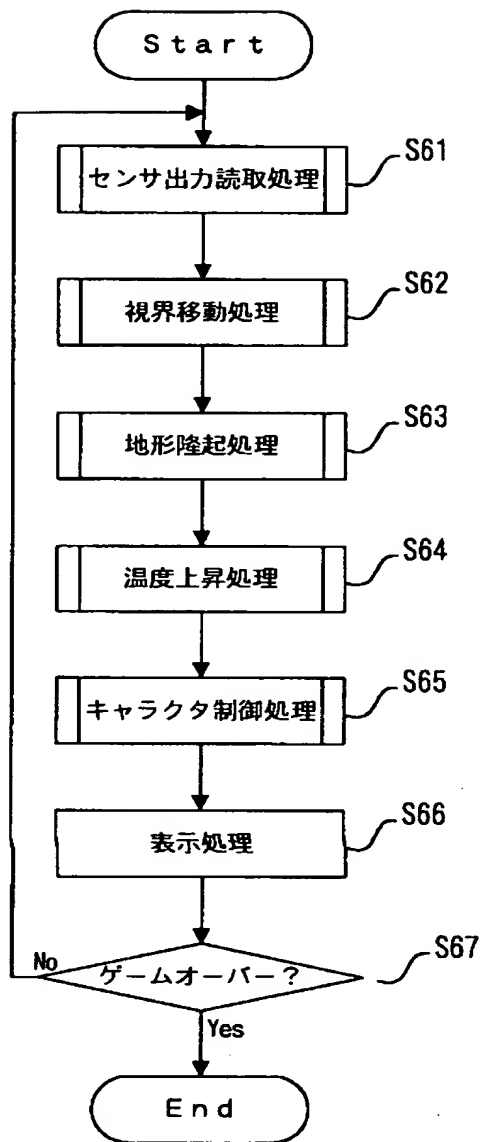
【図 4 5】



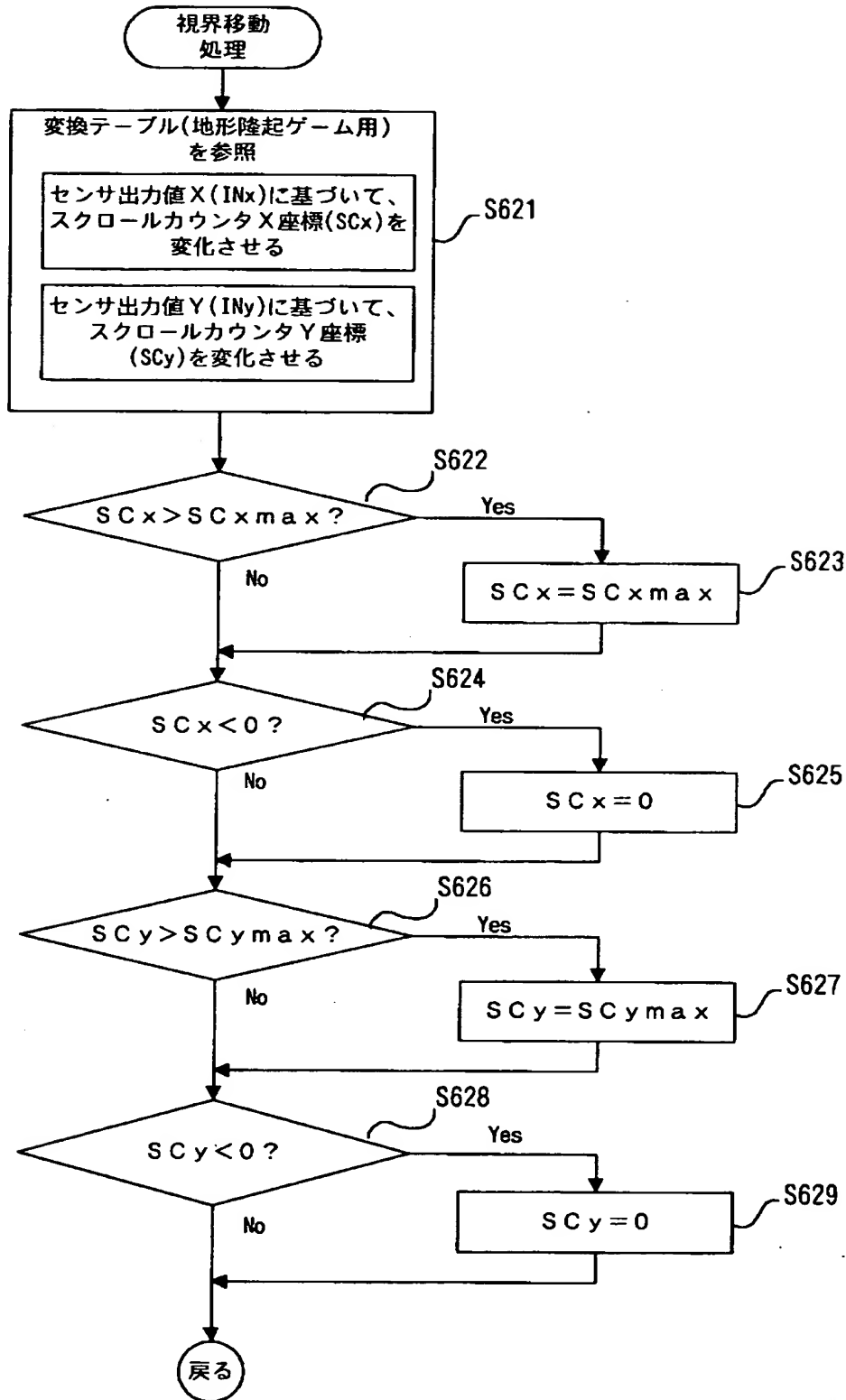
【図 4 6】



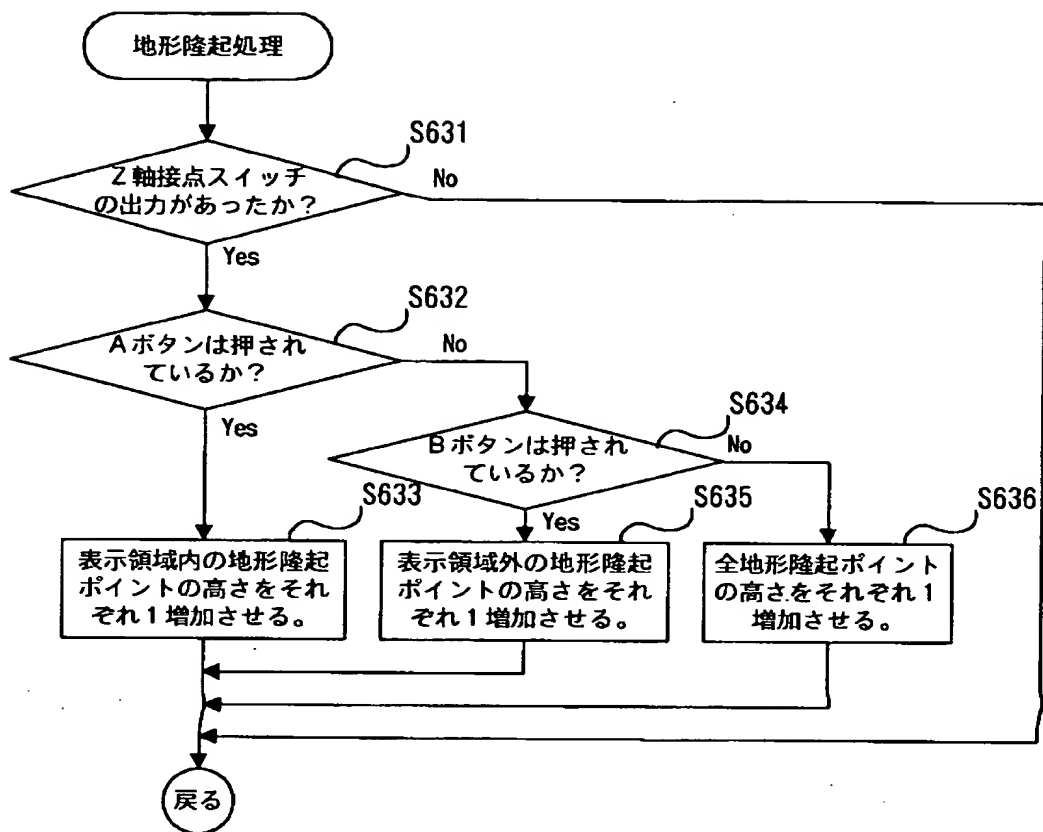
【図 4 7】



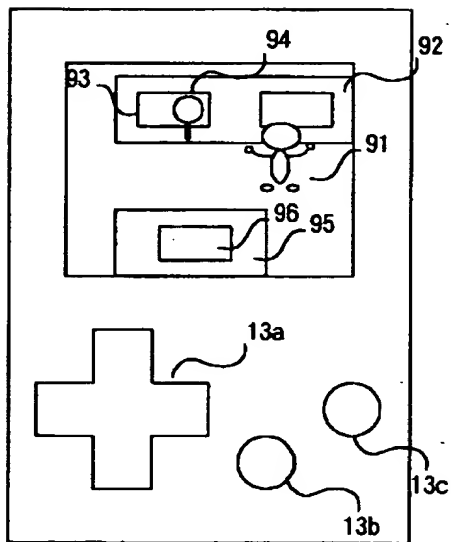
【図 48】



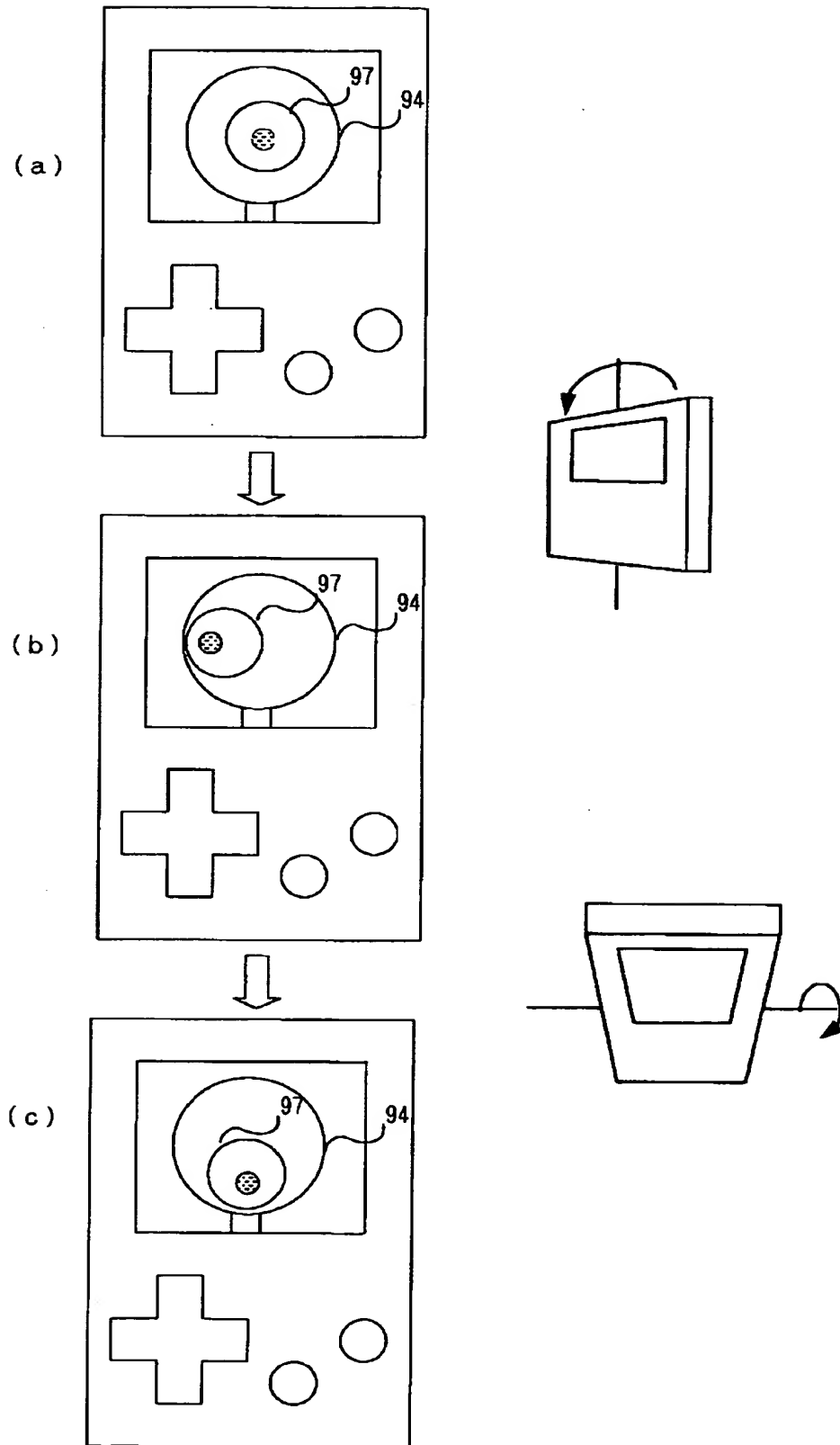
【図 4 9】



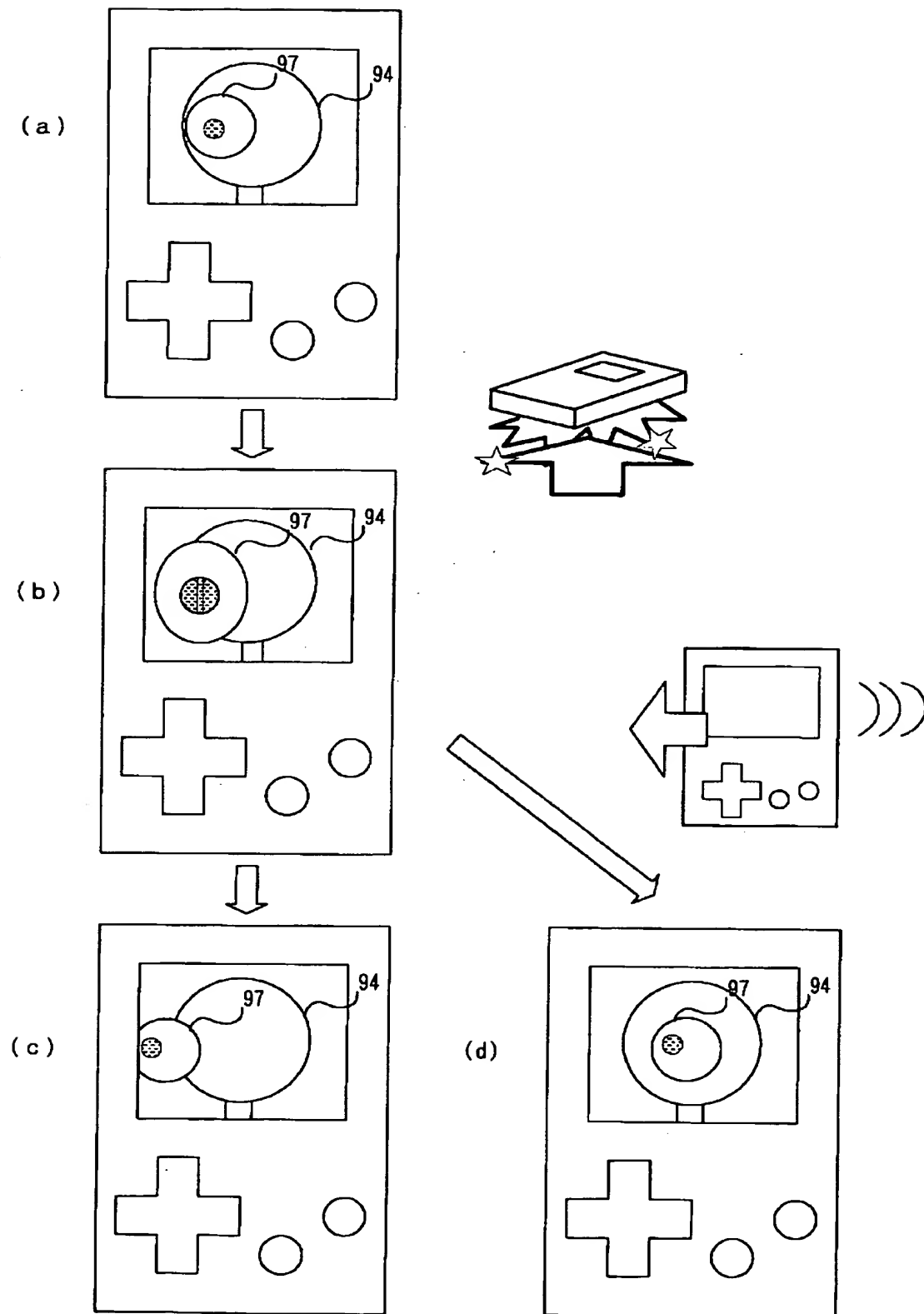
【図 5 0】



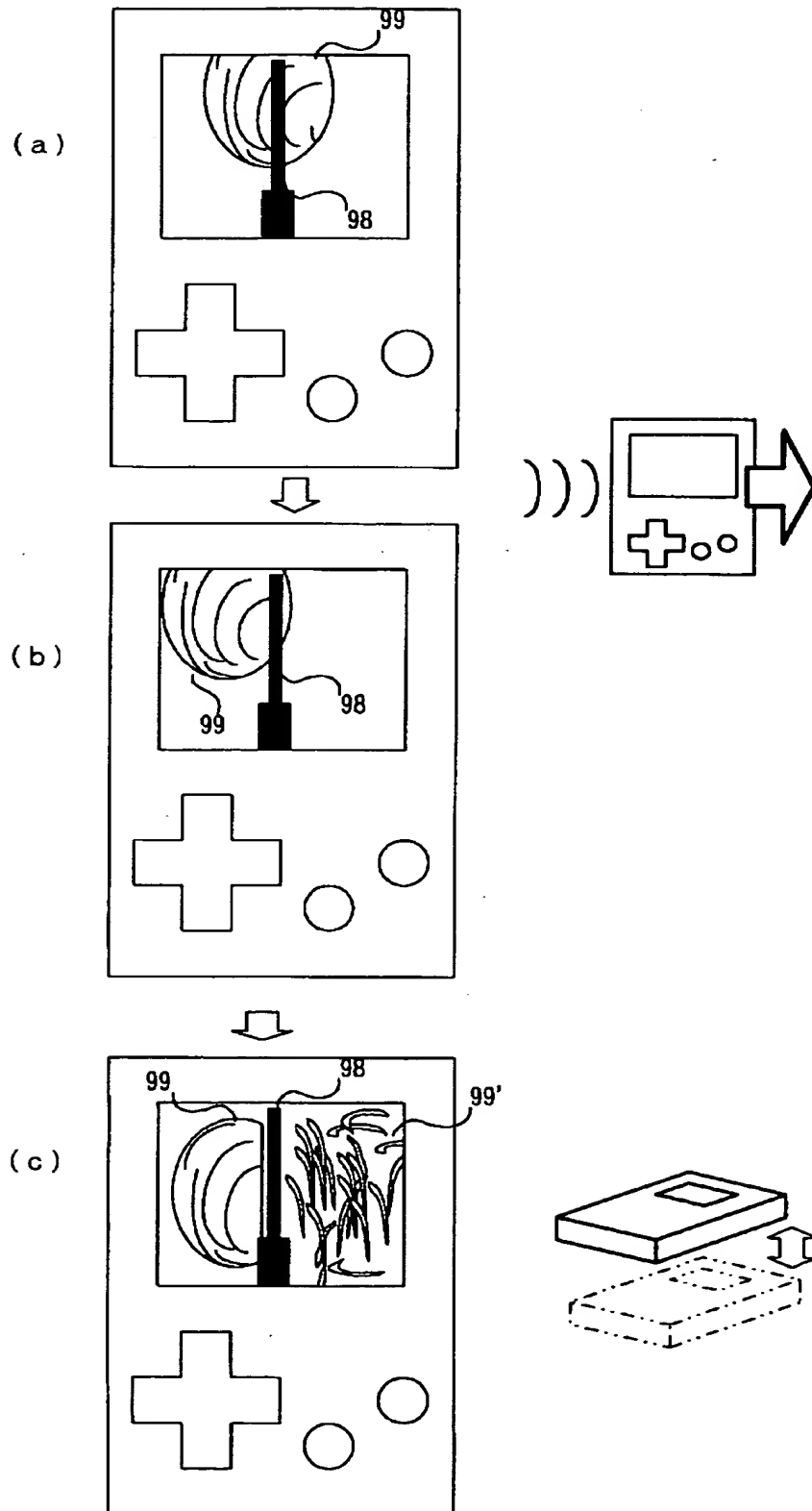
【図 5 1】



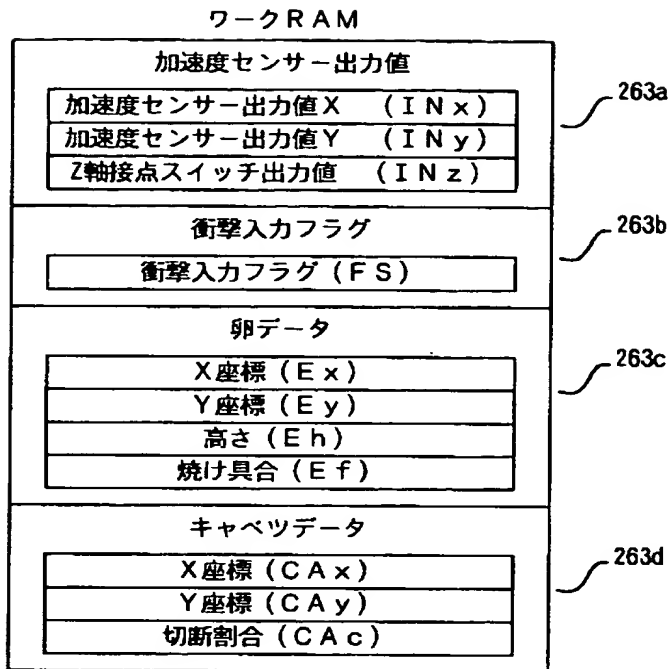
【図 52】



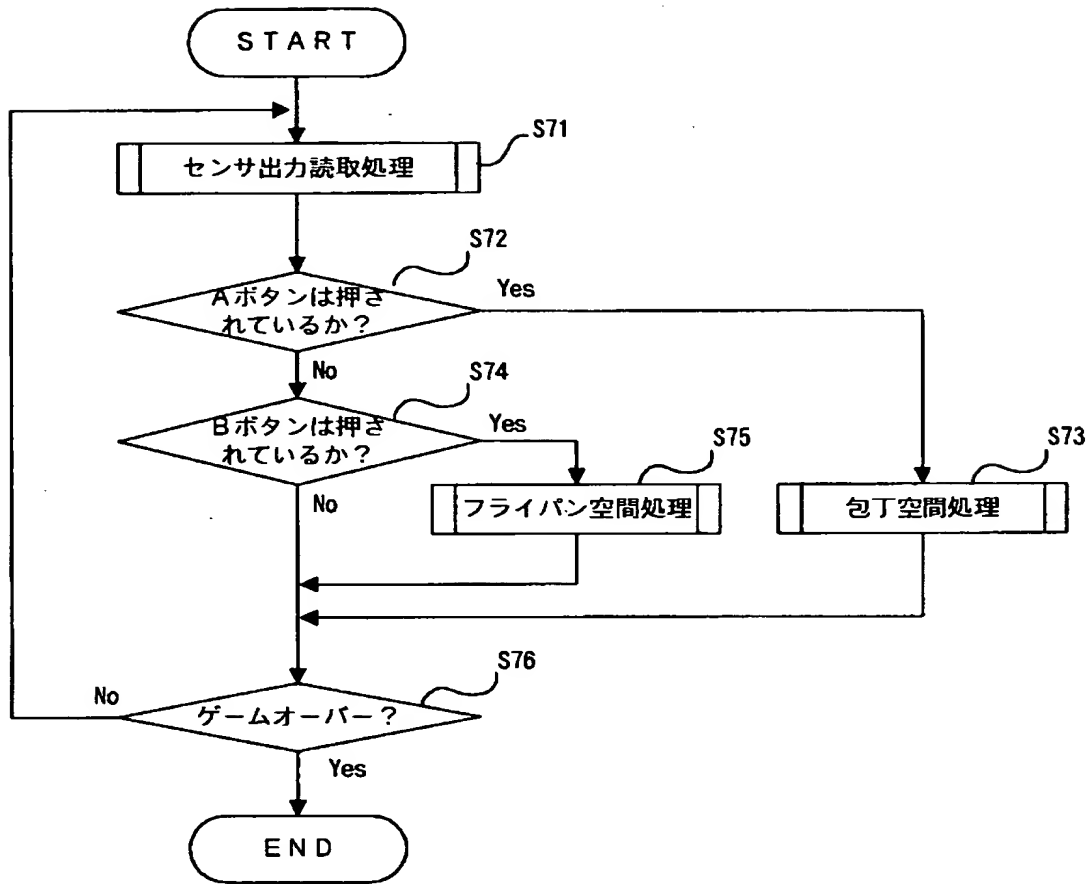
【図 5 3】



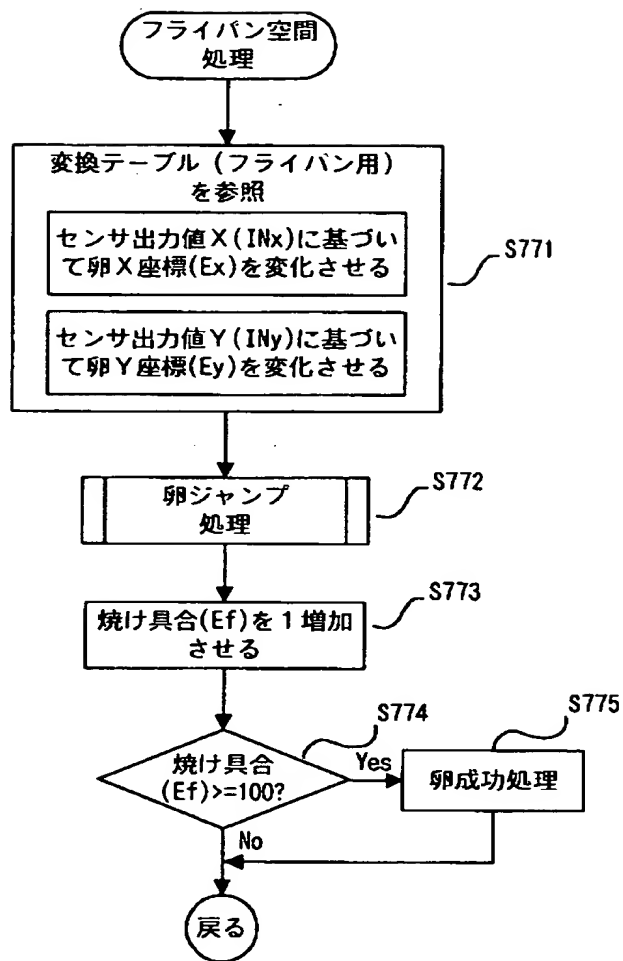
【図 5 4】



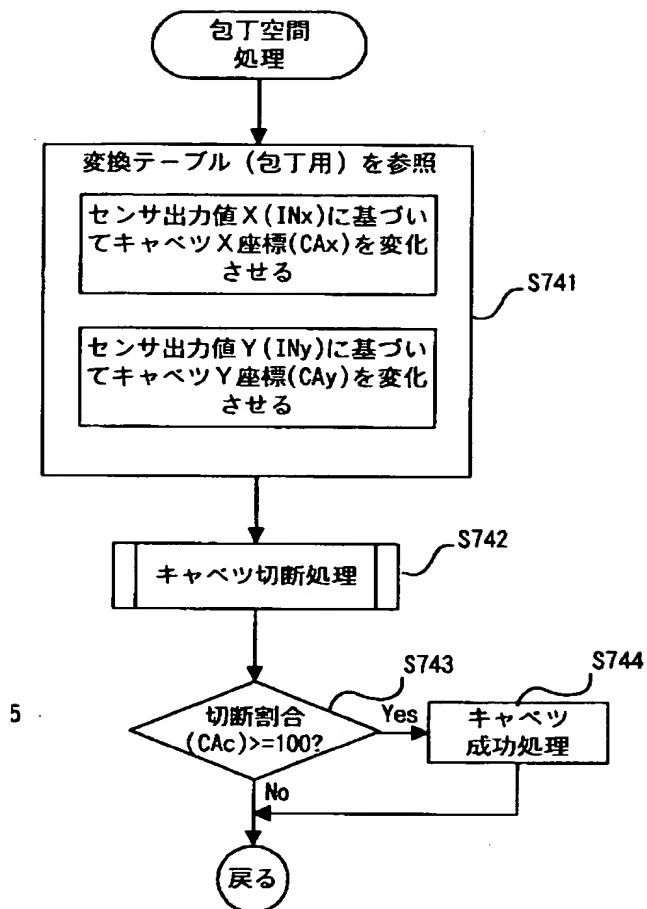
【図 55】



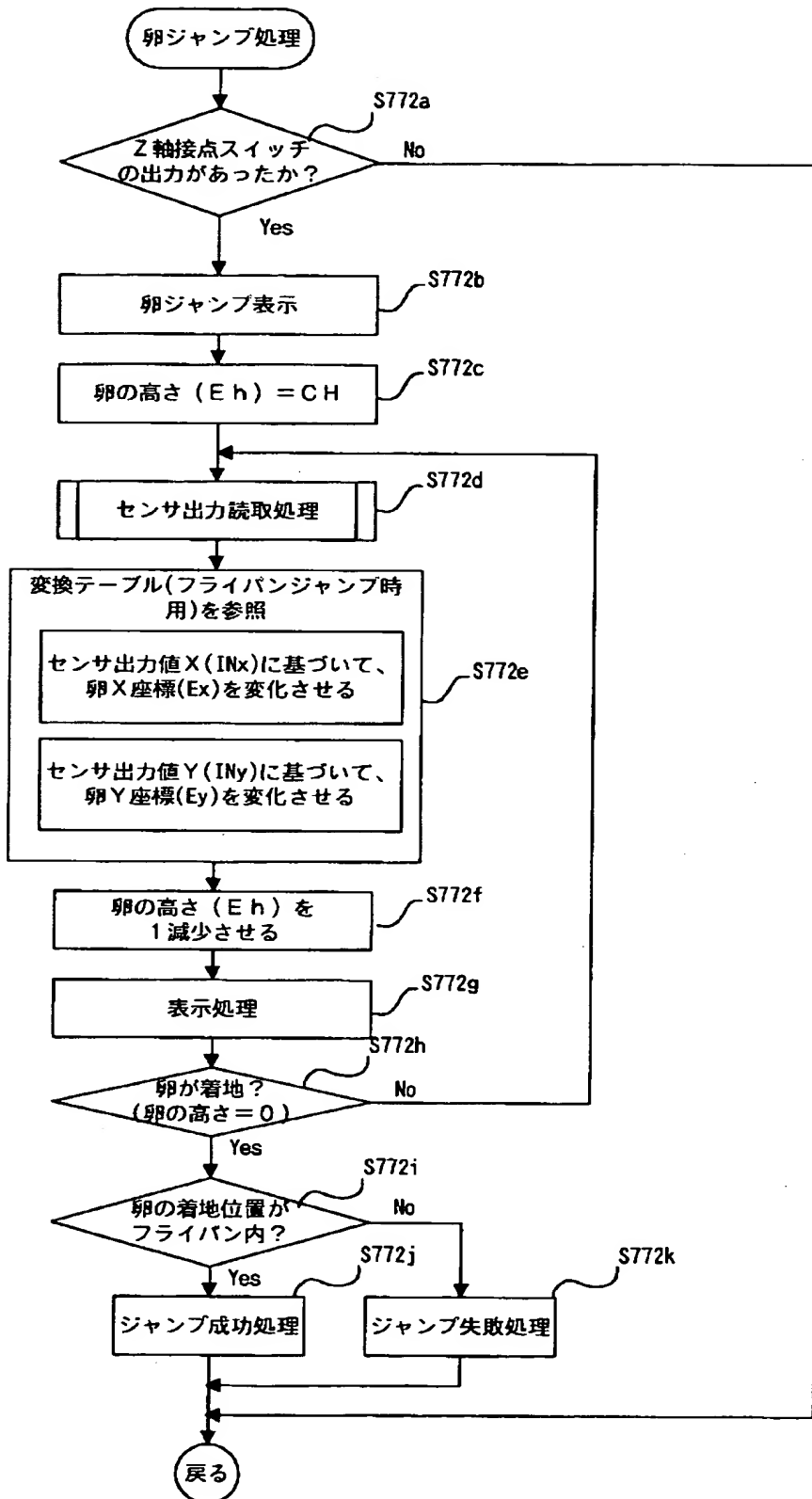
【図 56】



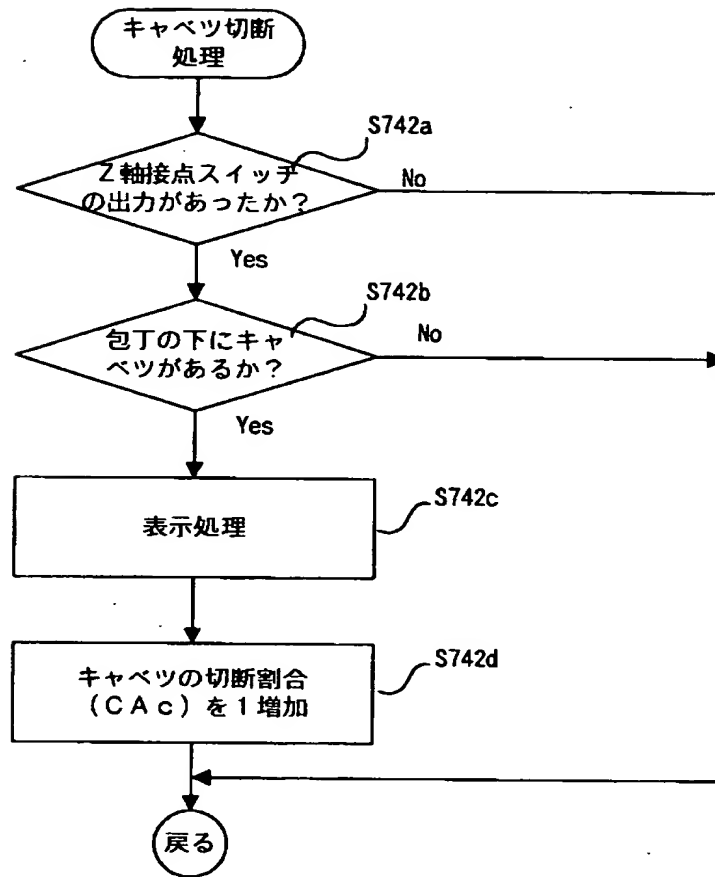
【図 5 7】



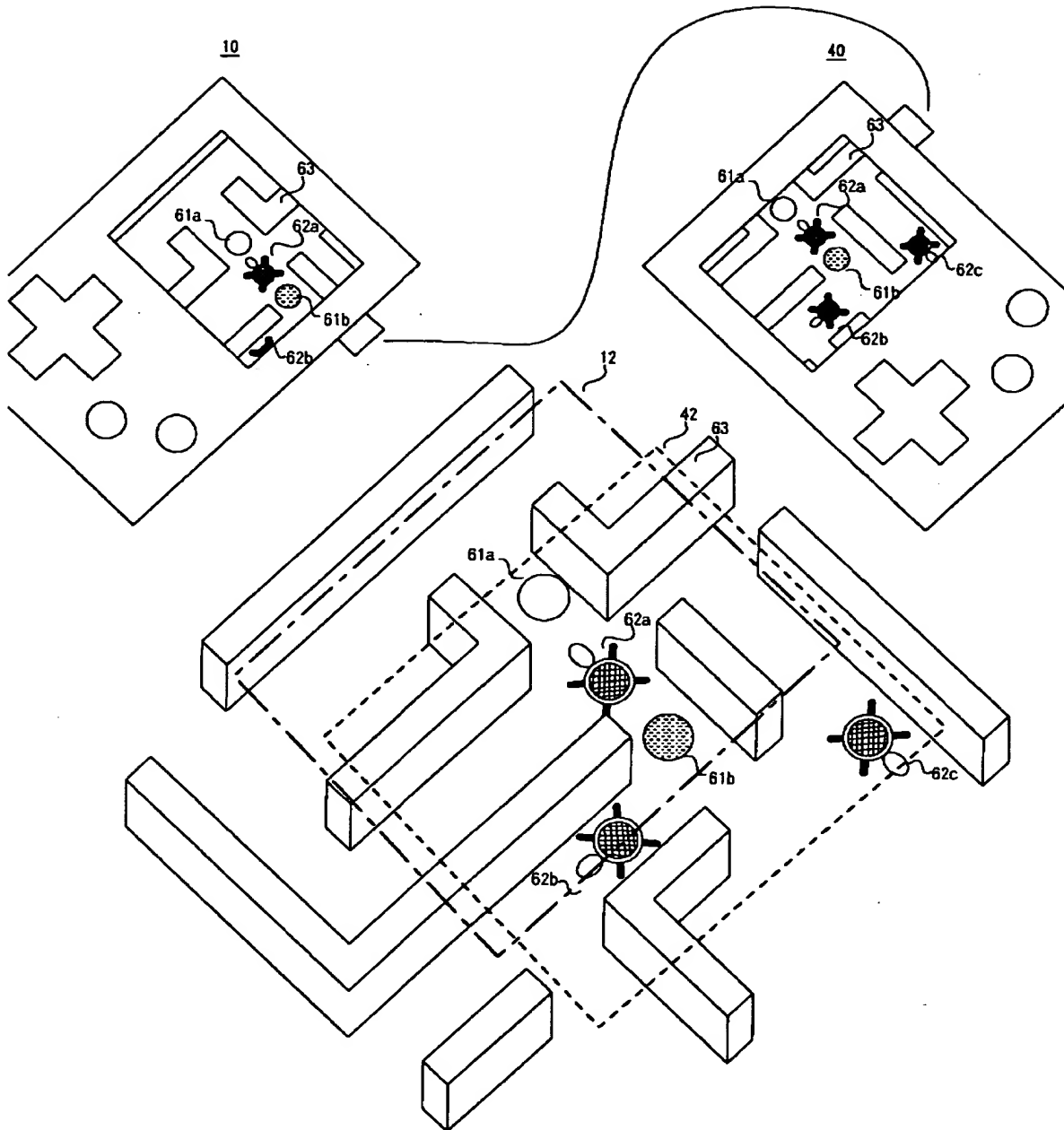
【図 58】



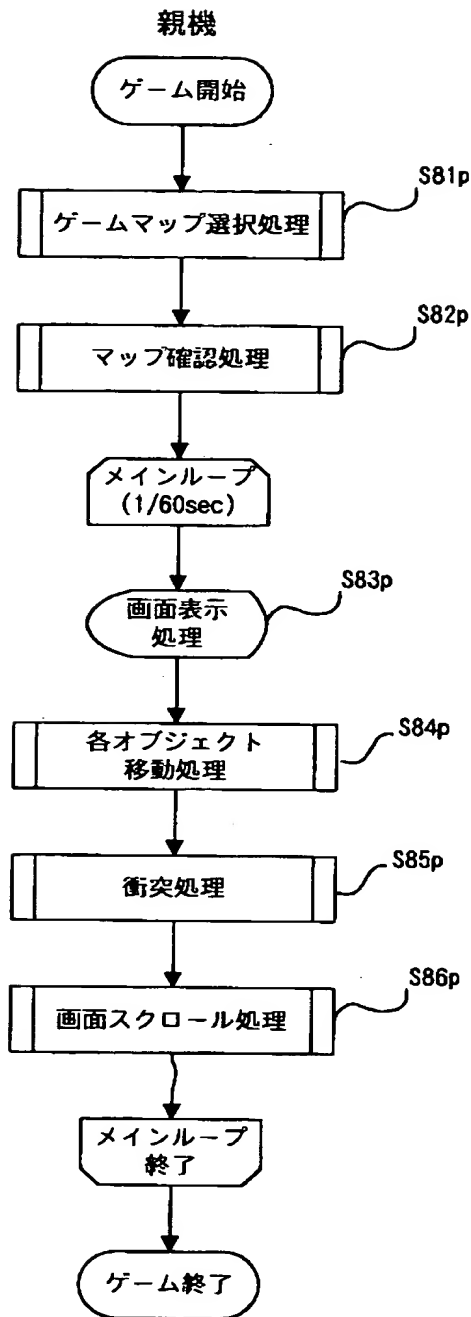
【図 5 9】



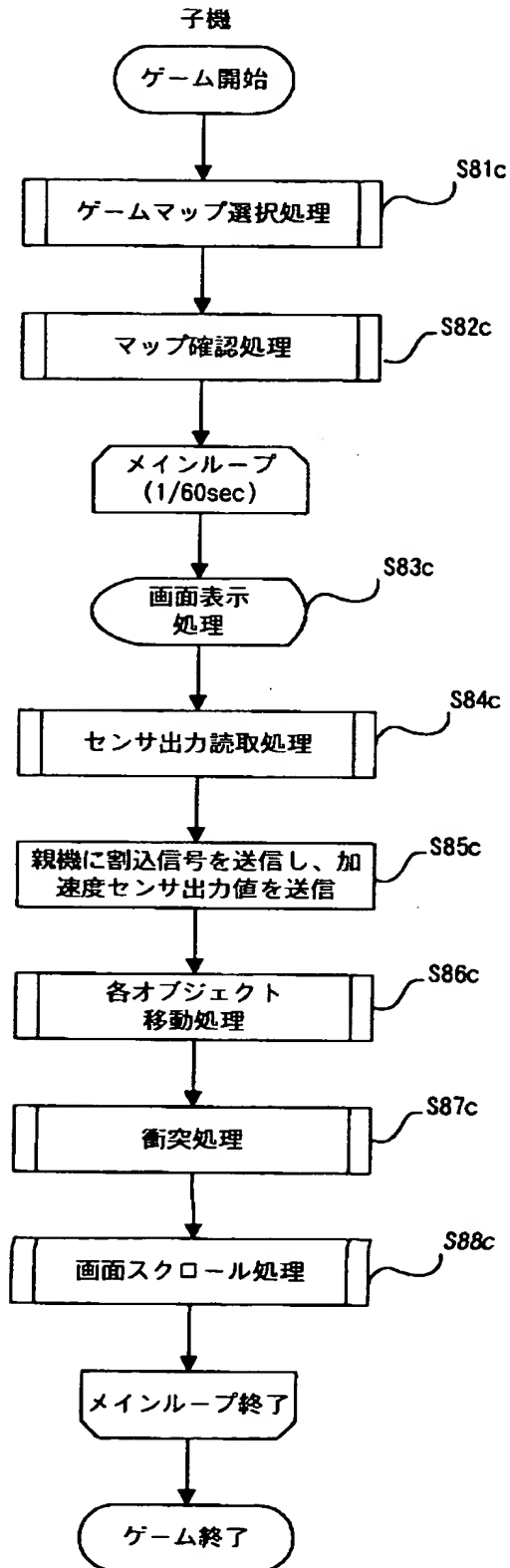
【図 60】



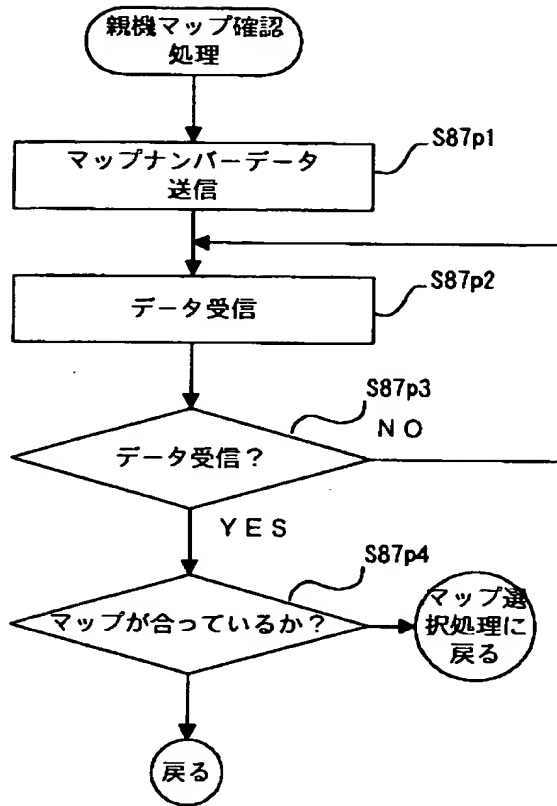
【図 6 1】



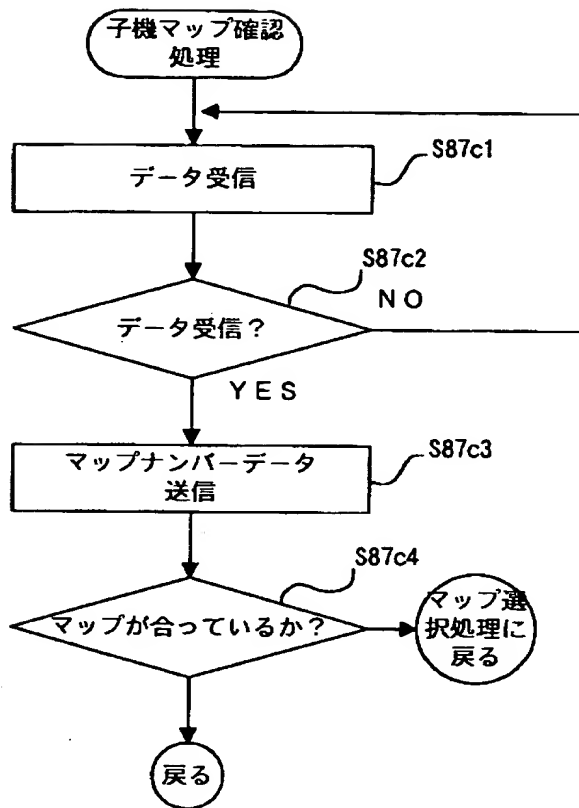
【図 6 2】



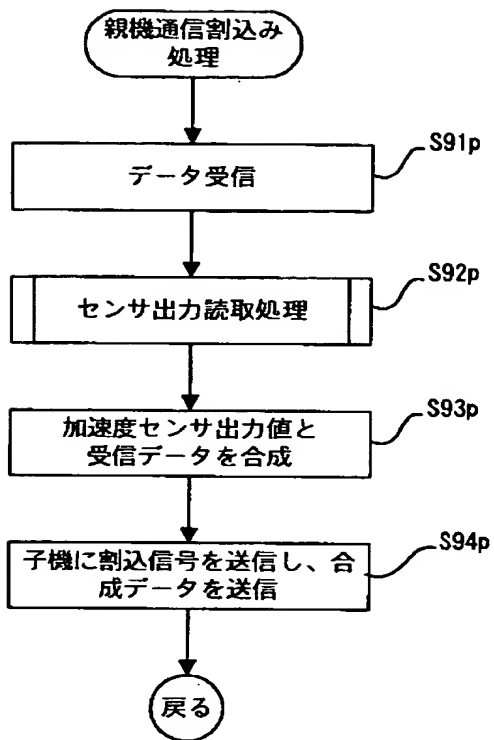
【図 6 3】



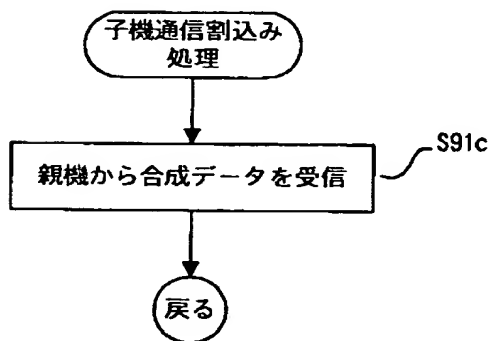
【図 6 4】



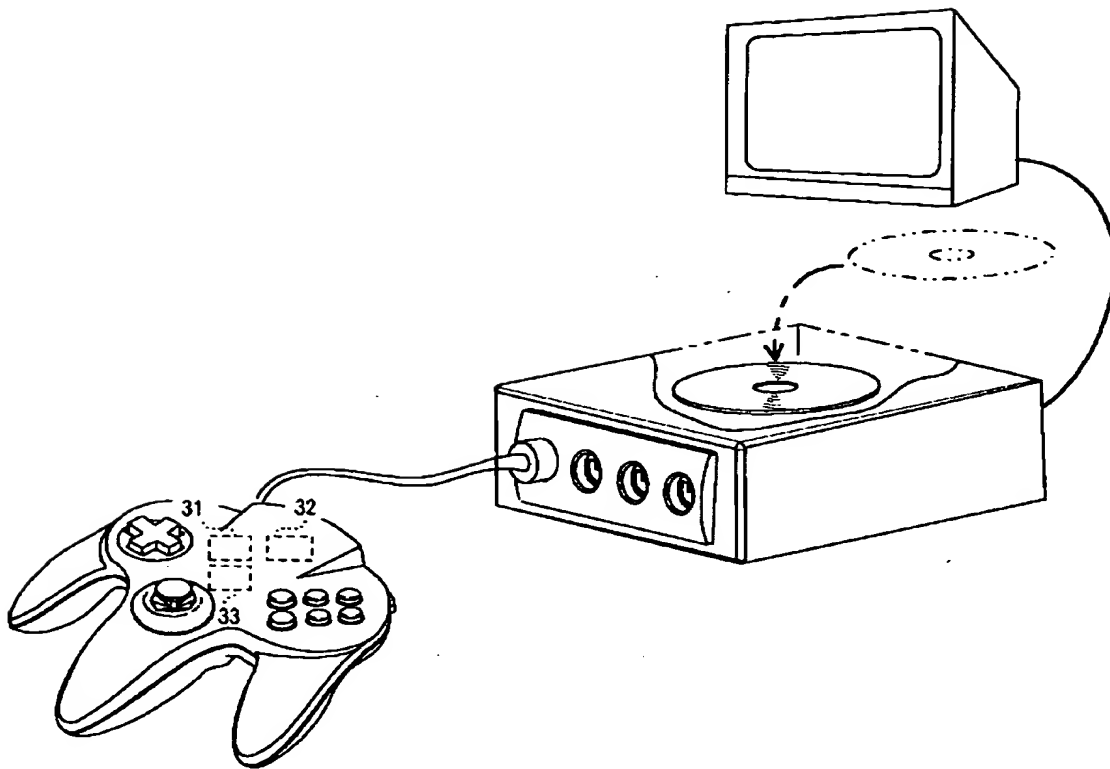
【図 6 5】



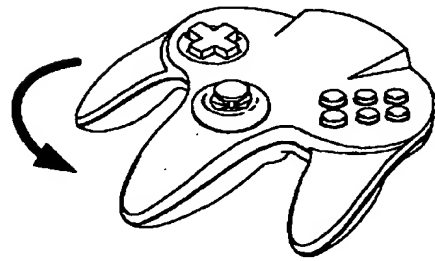
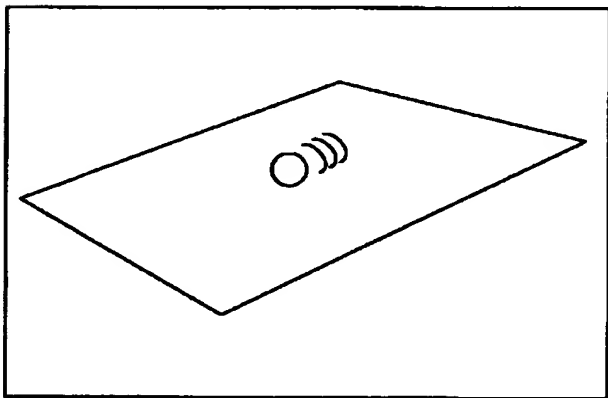
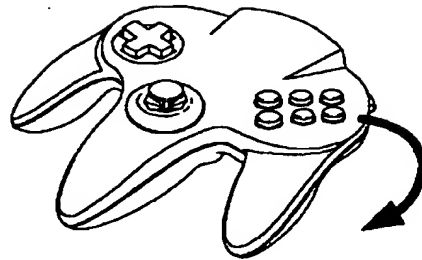
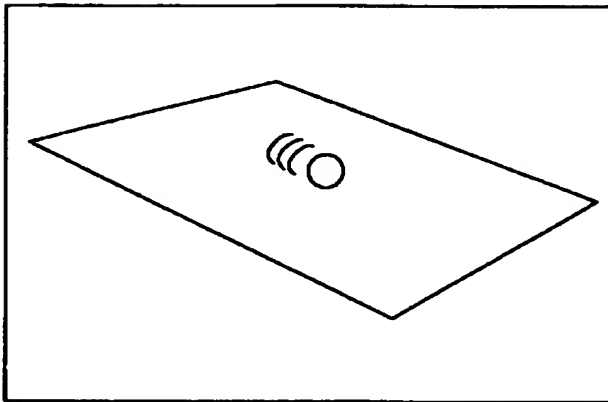
【図 6 6】



【図 67】



【図 68】



【書類名】 要約書

【要約】

【課題】

簡単な操作でゲーム空間の状態を変化させることが可能であり、プレイヤーが操作方法に熟練する必要がなく、ゲームプレイに集中できるとともに、熱中度を高められる、ゲームシステム及びそれに用いられるゲーム情報記憶媒体を提供する。

【解決手段】

プレイヤーによって把持されるハウジングに変化状態検出手段を備える。変化状態検出手段はハウジングに加えられた変化の量（例えば、傾き量，運動量，衝撃量等）と変化方向（例えば、傾き方向，運動方向，衝撃方向等）の少なくとも一方を検出する。シミュレーションプログラムが変化状態検出手段の出力に基づいて、ハウジングに加えられた変化量と変化方向の少なくとも一方に関連させて、ゲーム空間の状態を変化させるようにシミュレートする。

【選択図】 図 3

特 2000-174573

認定・付加情報

特許出願の番号	特願2000-174573
受付番号	50000722676
書類名	特許願
担当官	渡辺 正幸 7072
作成日	平成12年 6月14日

<認定情報・付加情報>

【提出日】 平成12年 6月 9日

次頁無

出 願 人 履 歴 情 報

識別番号 [000233778]

1. 変更年月日 1990年 9月 6日

[変更理由] 新規登録

住 所 京都府京都市東山区福稲上高松町60番地
氏 名 任天堂株式会社